**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**
**імені ІГОРЯ СІКОРСЬКОГО»**
**Інститут прикладного системного аналізу**

**Кафедра математичних методів системного аналізу**

«До захисту допущено»
В.О.Завідувача кафедри
_____ О.Л. Тимощук

# Дипломна робота

**на здобуття ступеня бакалавра**

**з напряму підготовки 6.040303 Системний аналіз**

**на тему: «Текстовий аналіз даних декларацій на предмет виявлення корупції»**

Виконав:
студент (-ка) IV курсу, групи КА-53
Якубець Андрій Олександрович _____

Керівник:
доцент, к.т.н.
Селін О. М. _____

Консультант з економічного розділу:
доцент, к.е.н.
Шевчук О. А. _____

Консультант з нормоконтролю:
доцент, к.т.н.
Коваленко А.Є. _____

Рецензент:

_____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2019 року

# РЕФЕРАТ

Дипломна робота: 120 с., 24 рис., 8 табл., 4 дод., 21 дж.

КОРУПЦІЯ, РЕГРЕСІЙНИЙ АНАЛІЗ, ПРОГНОЗУВАННЯ, МОДЕЛЮВАННЯ, ПАРСИНГ ДАНИХ, SAS.

Актуальність роботи – в Україні постійно відбуваються реформи у всіх сферах державного апарату. Одною із таких реформ – є антикорупційна реформа. Створення системи, яка зможе виявляти державних службовців за правилами ризику допоможе викорінити корупцію з найвищих ланок. Таким чином, розробка та карти ризику декларанта є актуальною на сьогоднішній день.

Об'єкт дослідження – щорічні декларації за 2017 рік розміщені у відкритому доступі на сайті Національного агентства с питань запобігання корупції.

Програмний продукт – реалізований за допомогою мови програмування SAS у середовищі розробки SAS ENTERPRISE GUIDE, а також мови програмування Python,

Мета роботи – створення програми для обробки великої кількості декларацій, що пришвидшить роботу антикорупційним агентствам для пошуку корупційних схем та прогнозування рівня корупції в Україні.

Метод дослідження – парсинг даних, розгляд та аналіз методів регресійного аналізу та дисперсійний аналіз.

За матеріалами бакалаврської роботи було опубліковано тези доповідей на конференціях:

- Всеукраїнська Інтернет-конференція здобувачів вищої освіти і молодих учених «Інформаційні технології: теорія і практика»;

Шляхи подальшого розвитку предмету дослідження – використання методів машинного навчання для автоматичного розпізнавання злочинних слідів.

# ABSTRACT

The theme: Text analysis of declarations for corruption detection.

Diploma work: 120 p., 24 fig., 8 tabl. 4 appendixes, 21 sources.

CORRUPTION, REGRESSION ANALYSIS, FORECASTING, MODELING, DATA PARSING, STATISTICAL ANALYTICAL SYSTEM.

Actuality – in Ukraine, reforms in all spheres of the state apparatus are constantly taking place. One of these reforms is the anti-corruption reform. Creating a system that can detect civil servants with abnormal earnings will help eliminate corruption from the highest levels. Thus, the development and scorecard of the declarant is relevant to date.

The object of study – Annual Declarations for 2017 are available on the website of the National Agency for the Prevention of Corruption.

The software product - implemented using Python and SAS programming language in the development environment of SAS ENTERPRISE GUIDE

Subject of research - predictive modeling techniques: regression models (including linear regression, ANOVA).

Purpose – creation of a program for handling a large number of declarations that will speed up the work of anti-corruption agencies to search for corruption schemes and predict the level of corruption in Ukraine.

The method of research – data parsing, review and analysis of regression analysis and correlation analysis.

Source material – bachelor research was published abstracts at conferences:

– All-Ukrainian Internet Conference of Higher Education Institutes and Young Scientists "Information Technologies: Theory and Practice".

The further development of the research subject – implementing of machine learning techniques, which automatically detect traces of crime.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ..... 9

ВСТУП ..... 10

РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ..... 11

1.1 Стан і тенденції корупції в Україні ..... 12

1.2 Про компанію SAS та її продукти в області аналітики та боротьби з корупцією ..... 14

1.2.1 SAS® Enterprise Guide ..... 15

1.2.2 Магічний квадрант для платформ розширеної аналітики станом на 2019 рік ..... 18

1.3 Загальний стан та основні показники корупції у світі ..... 20

1.3.1 Індекс сприйняття корупції ..... 21

1.3.2 Внутрішній валовий продукт країни ..... 22

1.3.3 Рівень безробіття ..... 22

1.3.4 Рівень інфляції ..... 23

1.4 Інформаційні технології в антикорупційних відомствах ..... 23

1.4.1 IBM® i2 Analyst's Notebook ..... 24

1.4.2 Qlik® Sense ..... 26

1.5 Постановка задачі дослідження ..... 28

Висновки до розділу 1 ..... 29

РОЗДІЛ 2 МАТЕМАТИЧНІ МЕТОДИ МОДЕЛЮВАННЯ, ЗНИЖЕННЯ РОЗМІРНОСТІ ТА ПОБУДОВИ КЛАСТЕРНИХ МОДЕЛЕЙ ..... 30

2.1 Регресійні моделі ..... 31

2.1.1 Модель авторегресії ..... 32

2.1.2 Модель ковзного середнього ..... 34

2.1.3 Лінійна регресія ..... 36

2.2 Краулинг інтернет сторінок ..... 38

2.2.1 Парсинг за допомогою мови програмування Python ..... 38
2.2.2 REST API ..... 40
2.2.3 Регулярні вирази ..... 41
2.3 Текстова аналітика ..... 42
2.4 Попередня обробка даних ..... 46
2.4.1 Обробка відсутніх значень ..... 48
2.4.2 Обробка екстремальних значень ..... 51
Висновки до розділу 2 ..... 53
РОЗДІЛ 3 ІНФОРМАЦІЙНО-АНАЛІТИЧНА СИСТЕМА АНАЛІЗУ МОДЕЛЮВАННЯ І ВИЯВЛЕННЯ КОРУПЦІЇ ..... 55
3.1 Опис реалізації ..... 55
3.2 Архітектура ..... 56
3.3 Технічні вимоги до системи ..... 57
3.4 Інструкція по експлуатації написаного програмного продукту ..... 57
3.4.1 Завантаження даних ..... 59
3.4.2 Класифікація декларацій за посадою та карта ризику ..... 62
3.4.3 Побудова прогнозуючої моделі ..... 64
3.5 Оцінка прогнозуючої моделі ..... 68
3.5.1 Дисперсійний аналіз ..... 69
3.5.2 Припущення щодо нормальності ..... 70
3.5.3 Тест на мультиколінеарність ..... 71
3.5.4 Тест на гетероскедастичність ..... 72
Висновки до розділу 3 ..... 74
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ ..... 76
4.1 Постановка задачі техніко-економічного аналізу ..... 77
4.1.1 Обґрунтування функцій програмного продукту ..... 78

4.1.2 Варіанти реалізації основних функцій ..... 78
4.2 Обґрунтування системи параметрів ПП ..... 80
4.2.1 Опис параметрів ..... 80
4.2.2 Кількісна оцінка параметрів ..... 81
4.2.3 Аналіз експертного оцінювання параметрів ..... 83
4.3 Аналіз рівня якості варіантів реалізації функцій ..... 87
4.4 Економічний аналіз варіантів розробки ПП ..... 88
4.5 Вибір кращого варіанта ПП за техніко-економічного рівня ..... 93
Висновки до розділу 4 ..... 93
ВИСНОВКИ ..... 95
ПЕРЕЛІК ПОСИЛАНЬ ..... 96
ДОДАТОК А ІЛЮСТРАТИВНІ МАТЕРІАЛИ ДО ДОПОВІДІ ..... 98
ДОДАТОК Б ТЕКСТ ПРОГРАМИ ..... 110
ДОДАТОК В ТЕЗИ КОНФЕРЕНЦІЇ БЕЗПЕКА СОЦІАЛЬНО-ЕКОНОМІЧНИХ ПРОЦЕСІВ В КІБЕРПРОСТОРІ ..... 121
ДОДАТОК Г ТАБЛИЦЯ СТАТИСТИЧНИХ ДАНИХ ..... 122

# ПЕРЕЛІК СКОРОЧЕНЬ

СППР – система підтримки прийняття рішень

АР – авторегресія

АРКС – авторегресія з ковзним середнім

ПЗ – програмне забезпечення

ПП – програмний продукт

ФВА – функціонально вартісний аналіз

SAS – Statistical Analysis System

БД – база даних США – Сполучені Штати Америки

ЄС – Європейський Союз

ANOVA – Analysis of Variance

CPI – Corruption Perceptions Index

IoT – Internet of Things

ВВП – Внутрішній валовий продукт країни

НАБУ – Національне антикорупційне бюро України

# ВСТУП

На сьогоднішній день, існує тенденція швидкого розвитку інформаційних технологій кожного дня. З цією тенденцією покращується життя населення, умови життя, а також з'являються нові технології для запобігання корупції. Зростаючий рівень хабарництва у державному секторі може придушити економічне зростання країни, а також позбавити її політичної свободи.

За законом про запобігання корупції від 14.10.2014 р. № 1700-VII. — всі державні службовці зобов’язані заповнити та оприлюднити декларації майнового стану для вільного доступу та їх перевірки. Починаючи з 2015 року, на сайті Національного агентства с питань запобігання корупції з’являється більше 1 млн. декларацій чиновників, суддів, прокурорів, присяжних, депутатів та кандидатів на посаду державного службовця. Попри таку велику кількість відкритих даних про майновий статус, у національних агентств основним інструментом виявлення корупціонерів є пошук незадекларованих статків, та вибіркова обробка декларацій найвищих категорій, котрі підозрюються у скоєнні корупційного злочину. Інші декларації залишаються неперевіреними, хоча й можуть містити сліди корупційного злочину.

Пояснювальна записка складається з чотирьох розділів. У першому розділі досліджуються та аналізуються існуючі підходи в боротьбі з корупцією, а також наявні інформаційні продукти. У другому розділі розглянуто ряд математичних моделей, за допомогою яких будують прогнози, методи обробки даних та викачування даних з мережі Інтернет. Третій розділ описує архітектуру розробленої програми, також у ньому аналізуються результати роботи алгоритму. Четвертий розділ присвячений функціонально-вартісному аналізу програмного продукта.

# РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

В цілому, корупція є формою нечесності або злочинної діяльності, яка здійснюється особою або організацією, на яку покладено владу, найчастіше для отримання незаконної вигоди. Корупція може включати в себе багато видів діяльності, включаючи хабарництво і розтрату, хоча вона також може включати в себе методи, які є законними в багатьох країнах. Політична корупція виникає, коли чиновник або інший державний службовець діє здійснювали свої офіційні повноваження для особистої вигоди [1]. Корупція є найбільш поширеним явищем в клептократії, олігархіях і мафіозних державах.

Корупція може відбуватися в різних масштабах. Корупція коливається від невеликих пільг між невеликим числом людей (дрібна корупція) до корупції, яка зачіпає уряд у великих масштабах (грандіозна корупція) і корупції, яка настільки поширена, що вона є частиною повсякденного структури суспільства.

Корупція і злочинність є ендемічними соціологічними явищами, які регулярно з'являються практично у всіх країнах в глобальному масштабі в різного ступеня і пропорції [2]. Кожна нація виділяє внутрішні ресурси для контролю і регулювання корупції і злочинності. Стратегії протидії корупції часто узагальнюються під загальним терміном «протидія корупції».

Окрім службовців, котрі виконують функції державного самоврядування, декларації мають подати такі категорії чиновників:

1) особи, що припиняють діяльність, пов'язану з виконанням функцій державного або місцевого самоврядування, подають декларацію особи, уповноваженої виконувати функції держави або місцевого самоврядування, на період, на який поширюються раніше подані заяви;

2) особи, які припинили діяльність, пов'язану з виконанням функцій органів державної влади або місцевого самоврядування, або інших видів діяльності, зобов'язані подати у наступному році після закінчення своєї діяльності. цієї статті - порядок декларування особи, уповноваженої виконувати функції держави або місцевого самоврядування за минулий рік [3];
3) особа, яка подає заяву на посаду, перед призначенням або обраний на відповідну посаду, подає у встановленому законом порядку особу, уповноважену виконувати функції державного або місцевого самоврядування протягом минулого року.

## 1.1 Стан і тенденції корупції в Україні

Корупція є широко поширеною проблемою в Україні. Це негативне явище, що системно обумовлює низький рівень розвитку економіки в державі, складність у залученні іноземних інвестицій, повільні темпи інтеграції з Європейським Союзом. Крім того, через те, що Державний бюджет регулярно не отримує значні кошти внаслідок недостатньої ефективності заходів із попередження та протидії корупції, соціальний захист державою уразливих груп населення знаходиться у вкрай незадовільному стані. В щорічному рейтингу Transparency International, Україна посіла 130 місце серед 180 країн за світовим Індексом сприйняття корупції (CPI). Таким чином, за винятком Росії, у рейтингу ТІ Україна стала найкорумпованішою країною Європи. У дослідженні Ernst&Young за 2017 рік Україна зайняла перше місце за поширеністю хабарництва/корупції у діловій практиці. Наприклад, принаймні 37% респондентів готові запропонувати грошову винагороду в обмін на укладення або продовження контракту [4].

Існує декілька видів корупції:

1) політична корупція – в роки після незалежності України було широко поширене шахрайство на виборах, в основному за рахунок використання «адміністративного ресурсу». Після фальсифікації результатів виборів 2004 року явна фальсифікація голосів зменшилася. Після цих виборів Верховний суд України постановив, що через масштаби фальсифікацій на виборах стало неможливо встановити результати виборів і розпорядився переглянути рішення. Хоча політики все ще стверджують, що фальсифікації виборів і адміністративні хитрощі, щоб отримати більше голосів за певну партію – не зникли. Український електорат як і раніше скептично ставиться до чесності виборчого процесу. Будь-який виборець, який бере участь у фальсифікаціях на виборах, може бути засуджений до позбавлення волі на строк до двох років, хоча активісти кажуть, що ще ніхто не був покараний за фальсифікацію виборців з моменту здобуття Україною незалежності;
2) місцева корупція – місцеве управління використовують свої пости для захисту своїх ділових інтересів;
3) юридична корупція – тиск на органи правосуддя, з метою винесення певного вироку. Судова система в Україні вважається корумпованою. Хоча судова незалежність існує в принципі, на практиці практично немає поділу на юридичні та політичні повноваження. Судді піддаються тиску з боку політичних і ділових інтересів;
4) корупція в державному секторі – привласнення коштів, призначених для ремонтування доріг, шкіл, медичних закладів і так далі. Основні обласні автомобільні дороги України, знаходяться в дуже поганому стані. Бюджет на ремонт доріг великий, але через корупцію бюджет розподіляється погано;
5) корупція у вищій освіті – студенти можуть «купити» вступ до коледжу, результати іспитів, отримати відзнаку за докторські або

магістерські дисертації. Деякі українські чиновники були спіймані з підробленими університетськими дипломами;

6) корупція в системі соціального забезпечення – українські ЗМІ опублікували безліч історій, що показують, що навіть парламентарії незаконно отримують соціальну допомогу, обманним шляхом стверджуючи, що вони ветерани війни та Чорнобиля;
7) корупція в охороні здоров'я – хоча медична допомога в державних лікарнях теоретично є безкоштовною для українців, пацієнти платять гроші там, щоб гарантувати, що вони отримують необхідне лікування, широко поширене.

## 1.2 Про компанію SAS та її продукти в області аналітики та боротьби з корупцією

SAS використовується у всьому світі приблизно в 118 країнах для вирішення складних бізнес-завдань. Більша частина програмного забезпечення керується за допомогою меню або через командний рядок. Як і інше програмне забезпечення, SAS має свою мову, яка може керувати програмою під час її виконання.

Професіонали в Статистичних Рішеннях, які є фахівцями в області програмного забезпечення для статистичного аналізу і допомагали тисячам кандидатів на докторантуру, магістрам і дослідникам.

SAS - це назва програмного забезпечення та назва компанії, яка її створила в 1970 році. У 1980 року було додано графіку, введення даних в режимі реального часу через Інтернет і вбудовано мову програмування С. У 1990-х роках SAS додав такі засоби, як візуалізація даних, адміністрування, зберігання сховищ даних і побудова інтерфейсів до Всесвітньої павутини та іншому [4].

SAS настільки потужний, що може обробити будь-які типи даних і може отримувати доступ до даних з будь-якого програмного забезпечення і будь-якого формату. Логічні операція також можуть бути виконані в SAS за допомогою операторів стандартних операторів. SAS виконує всі операції в циклі, крок за кроком, і виконує програму дуже швидко. Процедура ODS використовується для виведення результатів в інших форматах. Прикладами цього є HTML, RTF, Excel і так далі [5]. Також кожен може зробити макрос із програми SAS для задоволення різних дослідницьких потреб.

SAS дає вам можливість видобувати дані з різних джерел і аналізувати їх. Завдяки своїм потужним статистичним можливостям SAS може допомогти проаналізувати дані різними способами для задоволення різних потреб, наприклад:

1) багатовимірний аналіз;

2) бізнес-аналітика;

3) прогнозний аналіз;

4) створення безпечних ліків;

5) клінічні дослідження та прогнозування.

SAS також є мовою програмування четвертого покоління - тобто «мова програмування, розроблена з певною метою, наприклад, розробка комерційного програмного забезпечення для бізнесу». Він розроблений для зменшення зусиль програмування та мінімізації часу та витрат на розробку програмного забезпечення [6].

### 1.2.1 SAS® Enterprise Guide

SAS Enterprise Guide є потужним і інтуїтивно зрозумілим інструментом «drag and drop», що дозволяє інтегрувати і інтелектуальний аналіз даних, аналітику, звіти, експорт даних і багато іншого лише через один інструмент.

У SAS Enterprise Guide всі функціональні можливості включені у простий формат завдань. Завдання - це невеликі процеси, які виконують певні завдання, наприклад, перетинання двох таблиць може здійснюватися за допомогою Конструктора Запитів. Виконання графічного описового аналізу в SAS Enterprise Guide також являється простим завданням, яке може бути виконано за допомогою стандартних команд. Завдання очевидні і прості у використанні, вони не потребують знання програмування.

Це програмне забезпечення приносить всі можливості стандартної мови SAS в руки аналітика даних, не вимагаючи ніяких навичок програмування. SAS Enterprise Guide також може інтегруватися з додатковими інструментами, такими як Microsoft Excel, Word, PowerPoint і Outlook.

Також, програма містить:

1) конектори до різних технологій Великих даних, таких як Hortonworks, Cloudera, Pivotal та ін. Це дозволяє швидко і інтуїтивно отримувати більші вигоди від великих даних. Функції SAS: читання неструктурованих файлів (csv, logs, .txt, .xml тощо.), читання структуровані дані функції, через які можна скористатися перевагами Internet of Things в одному інструменті. Доступ до даних в Hadoop без написання жодного рядка коду;
2) інтегратор даних, через який ви можете мати єдиний вигляд даних в усіх середовищах з декількома базами даних, що є дуже поширеним у сьогоднішніх великих компаніях. У SAS Enterprise Guide будь-яке підключення до бази даних відбувається через стандарту бібліотеку, спосіб роботи з різними бібліотеками не змінюється, тому не потрібно окремо розбиратись з кожною базою даних;
3) інструменти для машинного навчання та інтелектуального аналізу даних, які дозволяють створювати якісні моделі за лічені хвилини;

4) інструменти для роботи з Microsoft Office, це означає, що все, що було зроблено в SAS Enterprise Guide, можна завантажити у ваш улюблений інструмент MS Office (Excel, Word, Power Point, Outlook).

Так, щомісячний звіт компанії може робитися автоматично, за допомогою SAS Enterprise Guide. Цей результат можна перенести до Microsoft Power Point лише за кілька кліків.

Компанія Gartner щорічно створює багато звітів, котрі мають назву «Магічний квадрант» та «Критичні можливості» для десятків видів IT-продуктів. Цей магічний квадрант оцінює постачальників платформ для аналізу даних та машинного навчання. Це програмні продукти, які дозволяють експертам з досліджень даних, звичайним аналітикам та розробникам додатків створювати, розгортати та керувати власними аналітичними моделями (рисунок 1.1).

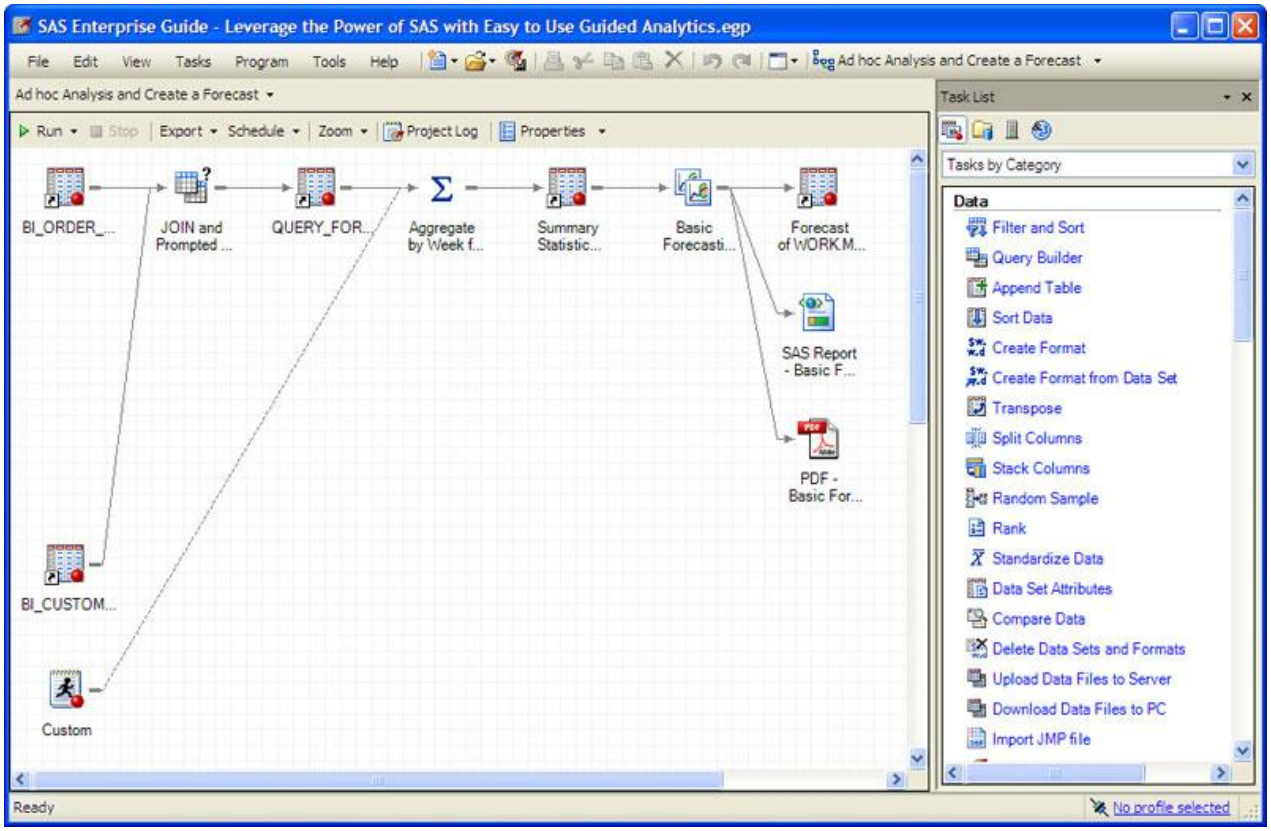



Рисунок 1.1 – Інтерфейс SAS® Enterprise Guide

### 1.2.2 Магічний квадрант для платформ розширеної аналітики станом на 2019 рік

Наукову платформа даних визначається як: Спільне програмне забезпечення, яке пропонує суміш основних програмних блоків, необхідних для створення всіх видів інформаційних рішень, і для включення цих рішень у бізнес-процеси, навколишню інфраструктуру та продукти.

Програмний продукт в основному використовує або поєднує різні пакети та бібліотеки - не вважається платформою для аналізу даних.

Не всі організації будують всю свою науку про дані та моделі ML з нуля. Деяким може знадобитися допомога, щоб розпочати або розширити свою наукову інформацію та ініціювати інвестування. Хоча цей Магічний квадрант оцінює доступність розфасованого контенту, такого як шаблони та зразки, він не оцінює постачальників послуг, які можуть допомогти прискорити або розширити науку про дані та застосування машинного навчання у всій організації. Також, він не оцінює спеціалізованих постачальників рішень, орієнтованих на галузь, домен або функцію. Найуспішніші компанії зображено на рисунку 1.2.

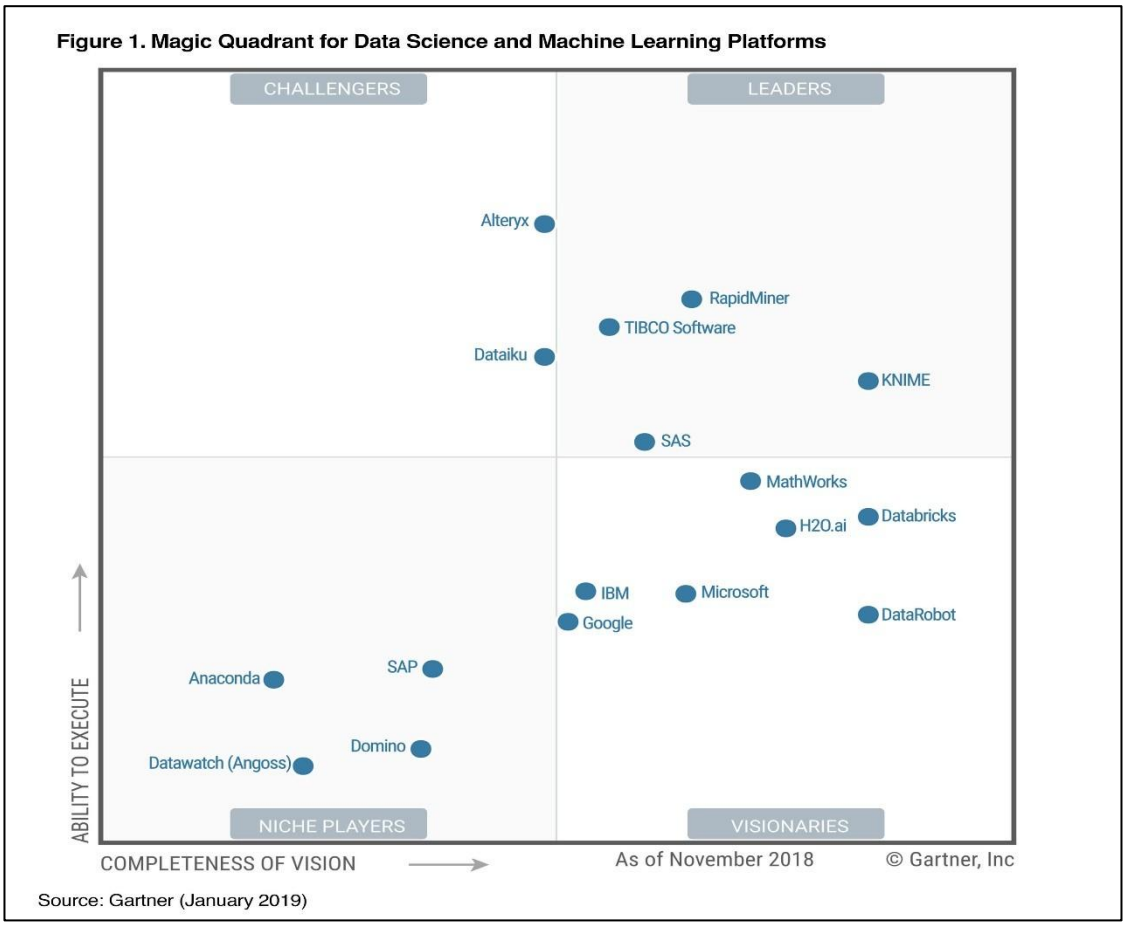



Рисунок 1.2 – Магічний квадрант для платформ розширеної аналітики (Advanced Analytics Platform)

Цей ринок має широкий спектр постачальників: Gartner запропонував широкому колу науковців даних та постачальників платформ машинного навчання взяти участь у процесі оцінки потенційного включення в цей магічний квадрант.

Користувачі цих платформ, які включають в себе вчених-дослідників даних, аналітиків та розробників додатків, котрі мають різні вимоги та переваги для користувацьких інтерфейсів (UI) та інструментів.

Фахівці-експерти вважають, що краще кодувати програми за допомогою Python або R. Іншим користувачам зручніше будувати моделі, використовуючи інтерфейс користувача через платформи.

Багато членів нових наукових спільнот даних підтримують набагато більш розширений підхід, який використовує методику «за лаштунками»,

щоб направляти цих менш досвідчених вчених-дослідників через процес побудови моделі та процесу операціоналізації.

З часом експерти з експертних даних можуть також віддати перевагу розширеному підходу, який дозволить їм ефективніше керуватися процесом побудови моделі та операціоналізації. Різноманітність прикладів та використання є більш важливою, ніж будь-коли.

Лідер може бути не найкращим вибором: широкий спектр доступних продуктів пропонує широкий спектр можливостей і різноманітні підходи до розробки, експлуатації та управління моделями. Тому важливо оцінити конкретні потреби при оцінці постачальників. Наприклад, постачальник у квадранті "Лідери" може бути не найкращим вибором.

Так само, нішовий гравець може стати ідеальним вибором. Проте, цей чарівний квадрант включає лише невеликий вибір сотень постачальників на цьому ринку.

## 1.3 Загальний стан та основні показники корупції у світі

Корупція, як було сказано раніше, є неминучим явищем для кожної країни. У зв'язку з цим важливо, щоб люди знали, що нинішня корупція впливає на зростання та розвиток країни, оцінюється шляхом досягнення макроекономічних цілей. Для подальшого аналізу цієї проблеми в даному дослідженні хотілося б встановити взаємозв'язок між економічним зростанням, ціновою стабільністю та повною зайнятістю.

В інтересах країни є можливість визначити, чи впливає корупція, виміряна Індексом сприйняття корупції, на досягнення макроекономічних цілей сталого економічного зростання, цінової стабільності та повної зайнятості. Для досягнення мети статті дослідницька робота також спрямована на те, щоб створити життєздатну економетричну модель,

керуючись конкретними економічними теоріями. Завдяки цьому, дослідження зможе сказати, що існує негативний вплив корупції на здатність країни досягати поставлених цілей для сукупної економіки.

Враховуючи той факт, що корупція є основною проблемою, яка є вродженою в кожній країні, необхідно вивчити її вплив на економічний сектор. Завдяки знанню наслідків, з якими країна страждає через корупцію в державному секторі, люди будуть поінформовані і будуть закликані до вжиття заходів щодо зменшення корупції.

### 1.3.1 Індекс сприйняття корупції

Індекс сприйняття корупції (CPI) оцінює країни щодо того, наскільки корумповані їхні уряди. Вона опублікована організацією Transparency International, яка прагне зупинити хабарництво та інші форми корупції в державі. Оцінка країни може варіюватися від нуля до 100, причому нуль вказує на високий рівень корупції та 100, що вказує на низький рівень. Transparency International запустив індекс у 1995 році, і сьогодні він нараховує 180 країн і територій.

Індекс сприйняття корупції вимірювався різними методологіями з року в рік, що робило важкі річні порівняння. Але в 2012 році методологія знову була змінена, на цей раз, щоб дозволити порівняння за часом.

За даними Transparency International, нова методологія передбачає чотири основні кроки, включаючи вибір вихідних даних, перерозподіл вихідних даних, агрегування перерозподілених даних і статистичних заходів, що вказують на рівень визначеності. У цей процес також включено механізм контролю якості. Це складається з незалежного збору даних та розрахунків двома власними дослідниками та двома незалежними дослідниками з наукових кіл.

### 1.3.2 Внутрішній валовий продукт країни

Внутрішній валовий продукт країни на одну особу отримується шляхом ділення її ВВП за певний період на його середнє населення за рік.

ВВП на душу населення є важливим показником економічних показників та корисним інструментом для порівняння середнього рівня життя та економічного добробуту між країнами. Проте ВВП на душу населення не є показником особистого доходу, а використання його для порівняння між країнами також має деякі слабкі сторони. Зокрема, ВВП на душу населення не враховує розподіл доходів у країні. Крім того, порівняння між країнами на основі долара США можуть бути спотворені коливаннями обмінного курсу і часто не відображають купівельну спроможність країн, які порівнюються.

### 1.3.3 Рівень безробіття

Рівень безробіття - це частка робочої сили, яка є безробітною, виражена у відсотках. Це відстаючий показник, що означає, що він, як правило, зростає або падає після зміни економічних умов, а не передбачає їх. Коли економіка знаходиться в поганому стані, а робочих місць мало, можна очікувати зростання безробіття. Коли економіка зростає на здорову ставку, а робочі місця відносно багато, можна очікувати падіння. Оскільки рівень безробіття має життєво важливу роль в економіці, необхідно спостерігати зв'язок між рівнем безробіття та корупцією.

Детермінанти корупції в економіці показують, що існує негативний зв'язок між рівнем безробіття та Індексом сприйняття корупції. Оскільки рівень безробіття також є неефективним в економіці, оскільки корупція є,

значимість рівня безробіття до корупції ще не визначена в цій роботі. Для розрахунку рівня безробіття береться відношення чисельності безробітного населення, розрахованого за методологією МОП, та економічно активного працездатного населення

### 1.3.4 Рівень інфляції

Інфляція являє собою зростання ціни в сукупній економіці. Збільшення ціни відповідає нижчій купівельній спроможності домогосподарств у країні, що в подальшому призведе до зниження сукупного випуску. Інфляція є остаточною корупцією, оскільки "інфляція викликана урядами, які постійно розширюють грошову масу країни, для якої вона має монополію", хоча центральний банк має право збільшувати або зменшувати грошову маса таким чином дозволяючи коливатися інфляції, уряд може мати владу над центральним банком через незаконні засоби.

## 1.4 Інформаційні технології в антикорупційних відомствах

Головним правоохоронним органом з питань корупції є Національне антикорупційне бюро України. Як сказано в пояснювальній записці до законопроекту, існуючий механізм протидії корупційним проявам в Україні неефективний.

Одним із шляхів підвищення ефективності протидії корупції є інституційна реформа органів, які здійснюють досудове розслідування і кримінальне переслідування у справах про корупційні злочини. Тому парламент схвалив створення нового автономного органу (поза системою

існуючих правоохоронних органів), основною функцією якого буде виявлення і розслідування корупційних злочинів, що становлять особливу суспільну небезпеку.

Завданням Національного бюро є протидія кримінальним корупційним правопорушенням, вчиненим вищими посадовими особами, уповноваженими на виконання функцій держави або місцевого самоврядування, і становлять загрозу національній безпеці. Подібні структури існують в США, Польщі, Франції, Сінгапурі, Ізраїлі та Індії.

НАБУ наразі використовує декілька аналітичних систем для своїх пошуків, котрі допомагають знаходити сліди корупційних злочинів. В наступних розділах приведені найбільш великі аналітичні платформи, що використовуються для пошуку зв’язків та різних схем.

### 1.4.1 IBM® i2 Analyst's Notebook

IBM i2 Analyst's Notebook - це інструмент візуального аналізу, який допомагає перетворити дані на розумне рішення. Рішення надає інноваційні функції, такі як візуалізації підключених мереж, аналіз соціальних мереж, геопросторові або часові види, які допомагають розкрити приховані з'єднання та шаблони даних. Ця інформація може допомогти вам краще виявити та зірвати злочинні, кібер-та шахрайські загрози. Цей інструмент підтримує такі функціональні можливості (рисунок 1.3)

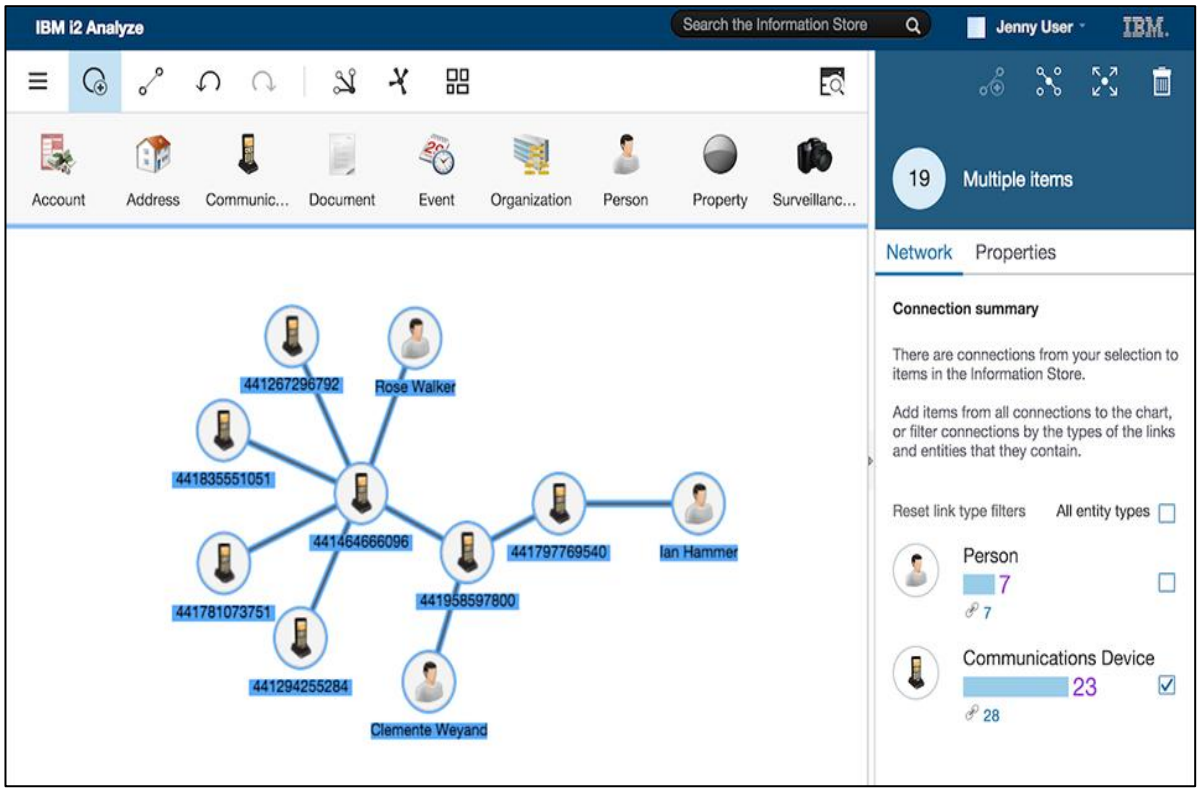



Рисунок 1.3 – Інтерфейс IBM® i2 Analyst's Notebook

1) візуалізація на вимогу – світ моделюється як сутності та зв'язки, шануючи спосіб навчання команд мислити. Це спрощує складні мережі, допомагає виявляти неочевидні структури і сприяє побічній думці;
2) кілька методів пошуку – збільшити потенціал пошуку інформації за допомогою декількох методів пошуку, включаючи текстовий пошук і візуальний запит; загальне середовище аналізу – IBM i2 Analyze дозволяє аналітикам і ширшим операційним командам працювати спільно через кордони, чи то організаційні чи географічні, обмінюючись інформацією і рішеннями в режимі реального часу;
3) гнучкий збір даних – доступ до широкого спектру різнорідних джерел даних для всебічного, ефективного та швидкого виробництва розвідувальної інформації. Залишати дані на джерелі або запобігати дані з декількох джерел в централізованому, агрегованому вигляді;

4) відкрита, розширювана архітектура – інтеграція з існуючою інфраструктурою для доповнення поточних процесів і процедур;
5) багатовимірна безпека – весь доступ контролюється за допомогою висококонфігурованої, дрібнозернистої і поширеної моделі безпеки. Використовуючи вбудовані інструменти або посилання на існуючі служби каталогів вашої організації, модель аутентифікації та керування користувачами забезпечує логічну сегментацію інформації, яка ґрунтується на оформленні, роботі, випадок або інших бізнес-вимогах.

### 1.4.2 Qlik® Sense

Qlik Sense - це платформа для бізнес-аналітики та візуальної аналітики, яка підтримує цілий ряд випадків використання, включаючи централізовано розгорнуті керовані аналітичні програми та панелі інструментів, спеціальні та вбудовані аналітики та візуалізацію самообслуговування, все в межах масштабованої, керованої структури. Рішення поставляється в трьох різних виданнях - Qlik Sense Desktop, Enterprise і Cloud.

Дану платформу використовує Prozzoro – системна реформа тендерного процесу в електронних публічних та державних закупівель в Україні. Сама система та авторські права на систему Prozorro «передані народу України (державі)» згідно з меморандумом.

Дані Prozzoro – є дозволеними для їх подальшого вільного використання та поширення. Будь-яка особа може вільно копіювати, публікувати, поширювати, використовувати, у тому числі в комерційних цілях, у поєднанні з іншою інформацією або шляхом включення до складу власного продукту, публічну інформацію у формі відкритих даних з

обов’язковим посиланням на джерело отримання такої інформації (рисунок 1.4).

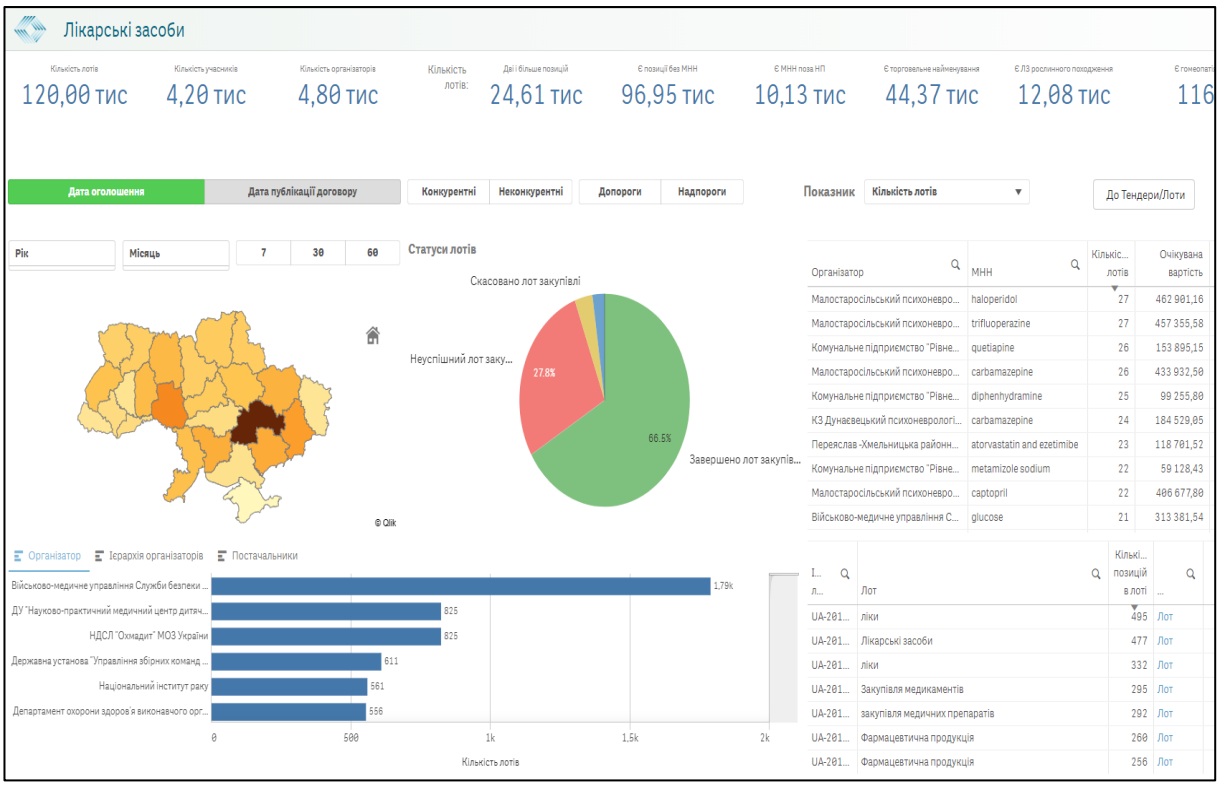



Рисунок 1.4 – Інтерфейс Qlik® Sense

Система Qlik Sense забезпечує візуалізацію даних і відкриття для окремих осіб і команд. Інструмент виявлення даних програмного забезпечення допомагає підприємствам будь-якого розміру досліджувати прості і складні дані і знаходити всі можливі асоціації в своїх наборах даних. За допомогою інтерфейсу перетягування користувачі можуть створювати інтерактивні візуалізації даних, щоб представити результат у формі історії.

Qlik Sense пропонує централізований вузол, з якого кожен користувач може обмінюватися даними та знаходити відповідні аналізи. Рішення здатне об'єднати дані з декількох баз даних, включаючи Cloudera Impala, IBM DB2, Microsoft SQL Server, Oracle, Sybase і Teradata. Open API також дозволяє розробникам впроваджувати Qlik Sense у нові програми та автоматизувати збір даних.

## 1.5 Постановка задачі дослідження

Метою бакалаврської роботи є дослідження існуючих методів аналізу та парсингу текстових даних відкритих даних Національного агентства с питань запобігання корупції, а саме декларацій держслужбовців. Також, попередня обробка даних, побудова таблиці значень ризиків декларанта та створення прогнозуючої моделі індексу корупції в Україні.

У рамках дипломної роботи потрібно:

1) розробити програму для парсингу, обробки та завантаження даних у інформаційно-аналітичну систему;
2) розробити структури таблиці значень ризиків декларанта, за заданими правилами ризику та критеріями відбору;
3) розробити програму для прогнозування корупції в Україні та провести аналіз на статистичну значимість результатів прогнозу;
4) протестувати програму на завантажених із сайту реальних даних.

Для вирішення поставлених задач, потрібно дослідити найпотужніші на сьогоднішній день алгоритми парсингу та прогнозування даних.

Об’єкт дослідження – щорічні декларації держслужбовців розміщені у відкритому доступі на сайті НАЗК, статистичні дані з держкомстату України та відкритих інтернет джерел.

Предмет дослідження – математичні методи побудови регресійних моделей, система пошуку та завантаження декларацій, аналітична платформа побудови правил ризику.

## Висновки до розділу 1

У розділі 1 дано визначення корупції як такої, описано стан і тенденції корупції в Україні. результати досліджень незалежного аналітичного агентства Gartner станом на лютий 2019 року для платформ розширеної аналітики у формі магічних квадрантів. За результатами цих досліджень компанію SAS визнано одним з лідерів в області платформ розширеної аналітики (Advanced Analytics Platform). На основі цього зроблено висновок, що для активної боротьби з корупцією необхідні інформаційні технології, які допомогли б знаходити її у можливих місцях.

Наведені основні макроекономічні показники країн, визначається їх поняття та основний зміст. Розглянуто Індекс сприйняття корупції, його методологію розрахунку та порядок розміщення країн у ньому.

Також розглянуто характеристику компанії SAS та IBM, їх стратегію та коло програмних рішень. Зокрема надано характеристики програмних продуктів: SAS® Enterprise Guide, Qlik® Sense та IBM® i2 Analyst's Notebook.

В кінці розділу наведено впровадженні аналітичні системи українських антикорупційних агентств НАБУ та Prozzoro.

# РОЗДІЛ 2 МАТЕМАТИЧНІ МЕТОДИ МОДЕЛЮВАННЯ, ЗНИЖЕННЯ РОЗМІРНОСТІ ТА ПОБУДОВИ КЛАСТЕРНИХ МОДЕЛЕЙ

Моделі описують наші переконання про функціонування світу. У математичному моделюванні ми перекладаємо ці переконання на мову математики. Це має багато переваг

1) Математика - дуже точна мова. Це допомагає нам формулювати ідеї та визначати основні припущення.
2) Математика - це лаконічна мова, з чітко визначеними правилами маніпуляцій.
3) У нашому розпорядженні всі результати, які математики довели протягом сотень років.
4) Комп'ютери можна використовувати для виконання чисельних розрахунків.
5) Комп'ютери можна використовувати для виконання складних алгоритмів.

У математичному моделюванні існує великий елемент компромісу. Більшість взаємодіючих систем у реальному світі є занадто складними для моделювання у всій їх повноті. Отже, першим рівнем компромісу є визначення найважливіших частин системи. Вони будуть включені в модель, решта буде виключена. Другий рівень компромісу стосується кількості математичних маніпуляцій, які варто. Хоча математика має потенціал довести загальні результати, ці результати суттєво залежать від форми використовуваних рівнянь. Невеликі зміни в структурі рівнянь можуть вимагати величезних змін у математичних методах. Використання комп'ютерів для обробки рівнянь моделі ніколи не може привести до елегантних результатів, але це набагато більш надійне проти змін.

## 2.1 Регресійні моделі

Регресійний аналіз є формою методики прогнозування, яка досліджує взаємозв'язок між залежною (цільовою) і незалежною змінною (предиктором). Ця методика використовується для прогнозування, моделювання часових рядів і пошуку взаємозв'язку причинного ефекту між змінними. Наприклад, взаємозв'язок між швидким рухом і кількістю дорожньо-транспортних пригод від водія краще вивчатися через регресію.

Регресійний аналіз широко використовується для прогнозування, де його практичний аспект має суттєве збіг з областю машинного навчання. Регресійний аналіз також використовується, щоб зрозуміти, які з незалежних змінних пов'язані з залежною змінною, а також для дослідження форми цих відносин.

У обмежених обставинах регресійний аналіз може бути використаний для виведення причинних зв’язків між незалежними і залежними змінними. Проте це може призвести до ілюзій або хибних стосунків, тому доцільна обережність.

Найвідоміші методи, такі як лінійна регресія і звичайна квадратична регресія, є параметричними, в яких функція регресії визначається через кінцеве число невідомих параметрів, які оцінюються з даних. Непараметрична регресія відноситься до методики, яка дозволяє функції регресії лежати в заданому наборі функцій, які можуть бути нескінченно-мірними.

Виконання методів на практиці залежить від форми процесу генерування даних, а також від того, як воно відноситься до використовуваного регресійного підходу. Оскільки справжня форма процесу формування даних взагалі не відома, регресійний аналіз часто залежить

певною мірою від припущень щодо цього процесу. Ці припущення іноді перевіряються, якщо наявна достатня кількість даних.

Регресійні моделі для прогнозування часто корисні навіть тоді, коли припущення помірно порушуються, хоча вони можуть не виконувати оптимально. Проте, у багатьох додатках, особливо при малих ефектах або питаннях причинності, заснованих на даних спостережень, регресійні методи можуть давати оманливі результати.

### 2.1.1 Модель авторегресії

Модель авторегресії (AR) являє собою випадковий процес, вона використовується для опису певних змін у природі, економіці – що змінюються в часі. Модель авторегресії вказує, що вихідна змінна залежить лінійно від власних попередніх значень і від стохастичності (недосконалості) [7].

Таким чином модель виглядає, як стохастичне різницеве рівняння. Разом з моделлю ковзного середнього (MA) це особливий випадок і ключовий компонент більш загальних моделей ARMA і ARIMA часових рядів, які мають більш складну стохастичну структуру; це також окремий випадок векторної авторегресійної моделі (VAR), що складається з системи більш ніж одного стохастичного різницевого рівняння. Позначення AR (p) вказує на авторегресивну модель порядку p. Модель AR (p) визначається як:

$$X_t = c + \sum_{i=1}^{p} a_i \cdot X_{t-i} + \varepsilon_t,$$

де $a_1,..a_p$ - параметри моделі (коефіцієнти авторегресії);

$c$ - константа;

$\varepsilon_t$ - білий шум.

Наприклад процес першого порядку – AR(1) є простою регресійною моделлю, в якій незалежна змінна просто відстає на один період:

$$X_t = c + r \cdot X_{t-1} + \varepsilon_t .$$

Для цього процесу коефіцієнт авторегресії співпадає з коефіцієнтом автокореляції першого порядку.

Рівняння з авторегресійною складовою має вигляд:

$$y(k) = a_0 + a_1 y(k-1) + a_2 y(k-2),$$

тобто, авторегресивна (AR) компонента другого порядку введена в рівняння регресії [8]. Щоб визначити, чи слід включити авторегресивну компоненту у рівняння регресії, необхідно обчислити та дослідити автокореляційну функцію змінної $y(k)$.

Коефіцієнти вибіркової автокореляційної функції розраховуються як:

$$r_y(s) = r_{y(k)y(k-s)} = \frac{1}{N-1} \frac{\sum_{k=s+1}^{N} \{[y(k) - y][y(k-s) - y]\}}{\sigma_y^2}, \quad s = 1,2,3,\ldots ,$$

де $\sigma_y^2$ – вибіркова дисперсія змінної $y(k)$.

Кількість коефіцієнтів АКФ, що не є нульовими в статистичному сенсі, вказують на порядок авторегресії.

Порядок компоненти авторегресії обчислюється за допомогою часткової автокореляційної функції (ЧАКФ):

$$\Phi_{11} = r(1),$$

$$\Phi_{22} = \frac{r_2 - r_1^2}{1 - r_1^2},$$

$$\Phi_{ss} = \frac{r_s - \sum_{j=1}^{s-1} \Phi_{s-1,j} r_{s-j}}{1 - \sum_{j=1}^{s-1} \Phi_{s-1,j} r_j}.$$

ЧАКФ чіткіше відображає порядок АР-моделі завдяки відсутності впливу проміжних коефіцієнтів кореляції на вибрані значення змінної. Тобто, коефіцієнт $\Phi_{11}$ характеризує степінь взаємозв'язку між сусідніми (в часі) значеннями змінної, а $\Phi_{22}$ характеризує взаємозв'язок між значеннями змінної, які розділені в часі двома періодами дискретизації [9].

### 2.1.2 Модель ковзного середнього

У статистиці, ковзне середнє – це розрахунок для аналізу точок даних шляхом створення серії середніх для різних підмножин повного набору даних. Варіанти включають в себе: прості і кумулятивні або зважені форми.

Враховуючи ряд чисел і фіксований розмір підмножини, перший елемент ковзної середньої величини отримують, взявши середнє значення початкового фіксованого підмножини ряду чисел.

Тоді підмножина модифікується шляхом "переміщення вперед"; тобто виключаючи перше число серії і включаючи наступне значення в підмножині. Загальна формула для зваженого ковзного середнього [10].

$$MA(k) = \frac{\sum_{i=1}^{N} w_i \cdot y(k-i+1)}{\sum_{i=1}^{N} w_i},$$

де $N$ – розмір вікна ковзного середнього;

$w_i$ – вагові коефіцієнти;

$y$ – часовий ряд вхідних даних.

Загальна формула для простого (арифметичного) ковзного середнього

$$MA(k) = \frac{\sum_{i=1}^{N} y(k-i+1)}{N}.$$

Тобто у випадку простого ковзного середнього всі вагові коефіцієнти мають однакову вагу (одиничну).

Експоненціальна ковзна середня є типом ковзної середньої, яка надає більшу вагу і значення найновішим точкам даних. Експоненціальне ковзне середнє значення також називають експоненціально зваженим ковзним середнім. Експоненціально зважена ковзна середня реагує більш суттєво на недавні зміни, ніж прості ковзні середні, які застосовують однакову вагу до всіх спостережень за період.

$$MA(i) = \frac{y(i-4) + y(i-3) + y(i-2) + y(i-1) + y(i)}{5}$$

$$MA(i) = \frac{y(i-2) + y(i-1) + y(i) + y(i+1) + y(i+2)}{5}$$

### 2.1.3 Лінійна регресія

Аналіз лінійної регресії використовується для прогнозування значення змінної на основі значення іншої змінної. Змінна, яку потрібно передбачити, називається залежною змінною. Змінна, яку ви використовуєте для прогнозування значення іншої змінної, називається незалежною змінною.

Ця форма аналізу оцінює коефіцієнти лінійного рівняння, що включають одну або більше незалежних змінних, які найкраще передбачають значення залежної змінної. Лінійна регресія відповідає прямій лінії або поверхні, що мінімізує розбіжності між прогнозованими та фактичними вихідними значеннями. Існують прості калькулятори лінійної регресії, які використовують метод "найменших квадратів" для виявлення найкращої лінії для набору парних даних. Потім оцінюється значення X (залежної змінної) від Y (незалежної змінної). Загалом лінійна регресійна модель визначається у виді:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_K x_K + u,$$

де $y$ — залежна пояснювана змінна;

$(x_1, x_2, \ldots, x_K)$ — незалежні пояснювальні змінні;

u — випадкова помилка, розподіл якої в загальному випадку залежить від незалежних змінних, але математичне очікування якої дорівнює нулю.

Відповідно, згідно з цією моделлю, математичне очікування залежної змінної є лінійною функцією незалежних змінних:

$$E\ y = \beta_0 + \beta_1 x_1 + \cdots + \beta_K x_K + u.$$

Вектор параметрів $(\beta_1, \beta_2, \dots, \beta_K)$ є невідомим і задача лінійної регресії полягає в оцінці цих параметрів на основі деяких експериментальних значень y і $(x_1, x_2, \dots, x_K)$. Тобто для деяких *n* експериментів є відомі значення $y_i, x_{i1}, \dots, x_{ip}\ _{i=1}^{n}$ незалежних змінних і відповідне їм значення залежної змінної.

Згідно з визначенням моделі для кожного експериментального випадку залежність між змінними визначається формулами:

$$y_i = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_K x_{K,i} + u_i,$$

або у матричних позначеннях $y = X\beta + u$, де:

$$y = \begin{matrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{matrix}, X = \begin{matrix} x_1' \\ x_2' \\ \vdots \\ x_n' \end{matrix} = \begin{matrix} 1 & x_{11} & \cdots & x_{1K} \\ 1 & x_{21} & \cdots & x_{2K} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \cdots & x_{nK} \end{matrix},$$

$$\beta = \begin{matrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{matrix}, u = \begin{matrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{matrix}.$$

На основі цих даних потрібно оцінити значення параметрів $(\beta_0, \beta_1, \dots, \beta_K)$, а також розподіл випадкової величини *u*. Враховуючи характеристики досліджуваних змінних, можна додати різні додаткові специфікації моделі та застосувати різні методи оцінки параметрів. Серед найбільш поширених специфікацій лінійних моделей є класична лінійна регресійна модель і узагальнена модель лінійної регресії.

## 2.2 Краулинг інтернет сторінок

Сканування зазвичай відноситься до роботи з великими наборами даних, де розробляються власні краулери (або боти), які сканують до веб-сторінок найнижчого рівня. Скрапери, з іншого боку, застосовуються для отримання інформації з будь-якого джерела (не обов'язково в мережі). Найчастіше, незалежно від підходів, ми посилаємося на добування даних з Інтернету як вискоблювання (або збирання), і це є серйозною помилкою.

Веб - це відкритий світ і типова практична платформа нашого права на свободу. Таким чином, багато контенту створюється, а потім дублюється. Наприклад, той самий блог може бути розміщений на різних сторінках, і наші краулери цього не розуміють. Отже, де-дублювання даних є невід'ємною частиною сканування даних. Це робиться для того, щоб досягти двох речей - зберегти клієнтів щасливими, не затоплюючи свої машини одними і тими самими даними більше одного разу, і зберігаючи власні сервери на деякому просторі.

Одна з найскладніших речей у просторі сканування в Інтернеті полягає в координації послідовних сканувань. Краулери повинні бути співвідносними з серверами, на які вони потрапляють. Протягом певного періоду часу, інтелектуальні краулери повинні стати більш розумними і навчитися знати, коли і скільки звернутись до серверу, щоб сканувати дані на веб-сторінках, дотримуючись політики ввічливості.

### 2.2.1 Парсинг за допомогою мови програмування Python

Дуже часто виникає необхідність витягти якусь інформацію з сайтів, в яких відсутній API. Доводиться застосовувати техніку, яка називається

парсинг. Парсинг - це аналіз вихідного HTML коду веб-сторінки і витяг необхідних шматочків інформації. Є два шляхи витягування потрібних відрізків інформації з HTML коду веб сторінок:

1) з використанням регулярних виразів;

2) з використанням спеціальних модулів.

Один з таких модулів - Beautiful Soup. Він дозволяє отримувати будь-які шматочки HTML коду і тексту (рисунок 2.1), роблячи вибірку на основі зазначених селекторів - класів або id шуканих тегів. У цьому він сильно схожий на JQuery.

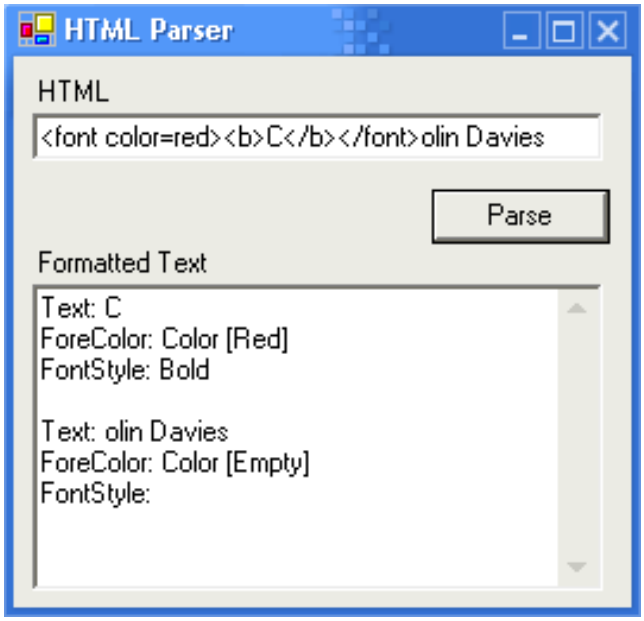


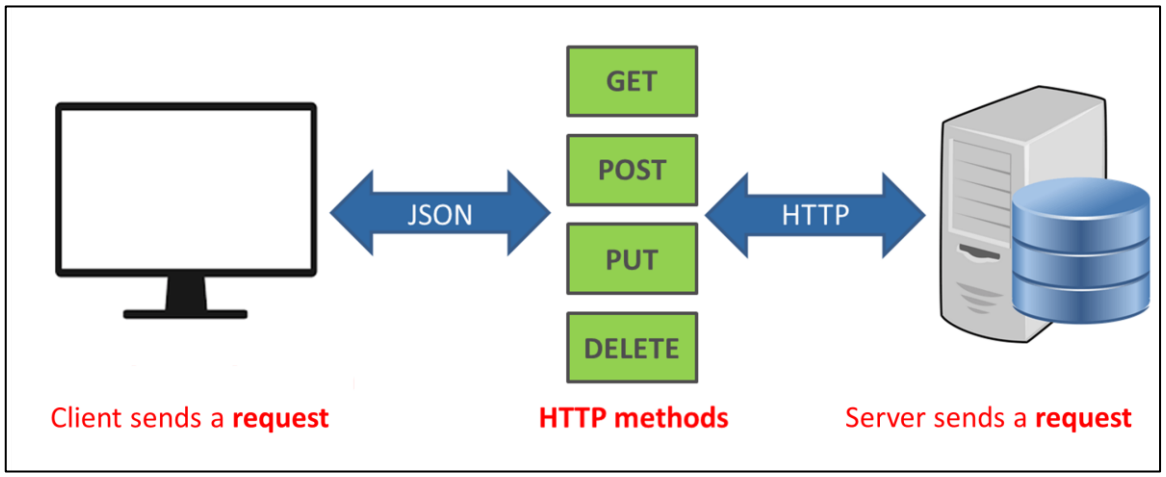

Рисунок 2.1 – Простий парсер HTML сторінок

Будь-яка веб сторінка представляє з себе набір HTML тегів - спеціальних слів, які укладені в дужки з значків більше-менше. Ми можемо отримати будь-який тег, звернувшись до нього по його класу. В результаті нам повернеться тег з шуканим id, або список з тегів, у яких є такий клас.

Щоб розпарсити якусь веб сторінку, потрібно відкрити її вихідний код, знайти в ньому потрібні нам теги, виписати їх класи або id, а потім звертатися до них по селекторам.

### 2.2.2 REST API

RESTful API - це інтерфейс прикладних програм, який використовує HTTP-запити до даних GET, PUT, POST і DELETE. Заснований на технології передачі державних даних, архітектурний стиль і підхід до комунікацій, які часто використовуються у розробці веб-служб, за принципом, зображеним на рисунку 2.2.




API

Рисунок 2.2 – Принцип роботи REST

Технологія REST, як правило, є кращою за технологію SOAP, оскільки REST використовує меншу пропускну здатність, що робить її більш придатною для використання в Інтернеті. API для веб-сайту - це код, який дозволяє взаємодіяти між двома програмами. API викладає правильний спосіб для розробника, щоб написати програму запиту служб з операційної системи або іншої програми. REST, що використовується браузерами, можна вважати мовою Інтернету. Зі зростанням використання хмар, з'являються API для розкриття веб-служб. REST - це логічний вибір для створення API, які дозволяють користувачам підключатися та взаємодіяти з хмарними службами. RESTful API використовуються такими сайтами, як Amazon, Google, LinkedIn і Twitter.

RESTful API розбиває транзакцію для створення серії невеликих модулів. Кожен модуль звертається до певної базової частини транзакції. Ця модульність надає розробникам велику гнучкість, але для розробників це може бути складним завданням. В даний час найпопулярнішими є моделі, що надаються Amazon Simple Storage Service, Cloud Data Interface та OpenStack Swift.

RESTful API явно використовує переваги методів HTTP, визначених протоколом RFC 2616. Вони використовують GET для отримання ресурсу; PUT для зміни стану або оновлення ресурсу, який може бути об'єктом, файлом або блоком; POST для створення цього ресурсу; і DELETE, щоб видалити його. За допомогою REST мережеві компоненти - це ресурс, до якого потрібно отримати доступ - чорна коробка, де деталі впровадження неясні.

Оскільки виклики не мають статусу, REST є корисним для хмарних додатків. Компоненти без статусу можуть бути вільно перерозподілені, якщо щось виходить з ладу, і вони можуть масштабуватися, щоб пристосувати зміни навантаження. Це відбувається тому, що будь-який запит може бути спрямований на будь-який екземпляр компонента; нічого не може бути збережене, яке потрібно запам'ятати наступною транзакцією. Це робить REST кращим для використання в Інтернеті, але модель RESTful також корисна для хмарних сервісів, оскільки прив'язка до сервісу через API - це питання контролю над тим, як URL декодується. Хмарні обчислення та мікросервіси майже напевно зроблять RESTful API дизайном правила в майбутньому.

### 2.2.3 Регулярні вирази

Регулярні вирази виникли в 1951 році, коли математик Стівен Коул Клені описав регулярні мови, використовуючи його математичні позначення, що називаються регулярними множинами. Вони виникли в теоретичній інформатиці, в підполях теорії автоматів (моделі обчислень) і в описі і класифікації формальних мов. Інші ранні реалізації відповідності шаблонів включають мову SNOBOL, який не використовував регулярні вирази, а замість цього власні конструкції відповідності шаблонів.

Регулярні вирази часто використовуються для позначення специфічного, стандартного текстового синтаксису (відмінного від математичного позначення) для представлення шаблонів по узгодженню тексту. Кожен символ у регулярному виразі, а саме кожен символ у рядку, що описує його шаблон – є або метасимволом, що має особливе значення, або звичайним символом, що має буквальне значення. Разом, метасимволи і літеральні символи можуть бути використані для ідентифікації тексту даного шаблону або для обробки ряду його примірників.

Шаблонові збіги можуть змінюватися від точного рівності до дуже загальної подібності, що контролюється метасимволами. Синтаксис метасимволів розроблений спеціально для представлення заданих цілей у стислій і гнучкої формі для автоматизації обробки тексту різноманітних вхідних даних у формі, яку легко вводити за допомогою стандартної клавіатури ASCII.

Звичайний контекст символів підстановки в глобусах подібних імен у списку файлів, тоді як регулярні вирази зазвичай використовуються в додатках, які узагальнюють текстові рядки.

## 2.3 Текстова аналітика

Інтелектуальний аналіз тексту - це процес вивчення та аналізу великих обсягів неструктурованих текстових даних, які можуть бути дослнза допомогою програмного забезпечення, яке може ідентифікувати концепції, шаблони, теми, ключові слова та інші атрибути в даних. Він також відомий як текстова аналітика, хоча деякі люди розрізняють два терміни; у такому вигляді аналітика тексту є додатком, що використовується за допомогою методів інтелектуального аналізу тексту для сортування наборів даних.

Текстова аналітика стала більш практичною для вчених даних та інших користувачів завдяки розробці великих платформ даних і глибоких алгоритмів навчання, які можуть аналізувати масивні набори неструктурованих даних.

Видобування та аналіз тексту допомагає організаціям знайти потенційно цінні бізнес-ідеї в корпоративних документах, електронних листах клієнтів, журналах контакт-центру, коментарі до опитувань, повідомлення соціальних мереж, медичні записи та інші джерела текстових даних. Можливості текстового видобутку все частіше включаються в чатботи та віртуальні агенти, які компанії розгортають для забезпечення автоматизованих відповідей клієнтам як частину своїх маркетингових, продажних і обслуговуючих операцій клієнтів.

### 2.3.1 Алгоритми класифікації

Задача класифікації може бути визначена як набір навчальних даних, що складається з записів. Кожен запис ідентифікується унікальним ідентифікатором запису і складається з полів, що відповідають атрибутам. Атрибут з неперервним доменом називається неперервним атрибутом.

Атрибут з кінцевим доменом дискретних значень називається категорійним атрибутом. Один з категорійних атрибутів - це

класифікаційний атрибут або клас, а значення в його домені називаються мітками класів.

Класифікація - це процес виявлення моделі для класу з точки зору інших атрибутів. Мета полягає в тому, щоб навчити набір навчальних даних для побудови моделі лейблу класу на основі інших атрибутів, так що модель може бути використана для класифікації нових даних не з набору даних навчання.

1) Метод кластеризації К-середніх є популярним алгоритмом аналізу даних, який спрямований на пошук груп у наборі даних. Число груп представлено змінною з назвою K. Це один з найпростіших алгоритмів навчання, який розв'язує проблему кластеризації. Ключова ідея полягає у визначенні k центроїдів, які використовуються для позначення нових даних. Кластеризація К-середніх - класичний спосіб класифікації тексту. Він широко використовується для класифікації документів, створення кластерів у текстових даних соціальних медіа, ключових слів для пошуку в кластерах тощо.
2) Простий класифікатор Байєса вважається одним з найбільш ефективних алгоритмів інтелектуального аналізу даних. Це простий імовірнісний алгоритм для задач класифікації. Класифікатор байєса заснований на так званій байєсовській теоремі і дає чудові і надійні результати, коли він використовується для аналітики текстових даних. Він не є єдиним алгоритмом, а являє собою сімейство алгоритмів, які припускають, що значення ознак, що використовуються в класифікації, є незалежними.

### 2.3.2 Алгоритми виявлення асоціацій

Щоб виявити асоціації, присутні в даних. Проблема була сформульована спочатку в контексті даних транзакцій у супермаркеті. Дані цього кошика, як відомо, складаються з операцій, здійснених кожним клієнтом. Кожна транзакція містить товари, придбані замовником. Мета полягає в тому, щоб переконатися, чи можна використовувати виникнення певних елементів у транзакції, щоб вивести виникнення інших елементів, або, іншими словами, знайти асоціативні відносини між елементами. Традиційно моделі асоціацій використовуються для виявлення бізнес-тенденцій шляхом аналізу операцій клієнтів. Однак вони також можуть ефективно використовуватися для прогнозування доступу до веб-сторінки для персоналізації.

1) Паралельний алгоритм виявлення асоціацій: Проблема може бути викладена як задана сукупність елементів, правила асоціації передбачають виникнення іншого набору елементів з певним ступенем довіри. Мета полягає в тому, щоб виявити всі такі цікаві правила. Існує декілька властивостей моделей асоціацій, які можна обчислити.
2) Послідовний алгоритм пошуку асоціації: концепція правил асоціації може бути узагальнена і більш корисною, спостерігаючи інший факт про транзакції. Всі транзакції мають тимчасову мітку, пов'язану з ними; тобто час, коли відбулася операція. Якщо цю інформацію використовувати, можна знайти такі відносини, як наприклад, якщо клієнт купив книгу сьогодні, то він, ймовірно, купить книгу і через кілька днів. Корисність такого роду правил породила проблему виявлення послідовних моделей або послідовних асоціацій. Загалом, послідовний шаблон є послідовністю наборів елементів з різними обмеженнями часу, що накладаються на входження елементів, що з'являються в шаблоні.

### 2.3.3 Алгоритми кластеризації

Кластеризація - це поділ даних на групи подібних об'єктів. Кожна група, що називається кластером, складається з об'єктів, які подібні між собою і не схожі на об'єкти інших груп. Представлення даних меншою кількістю кластерів обов'язково втрачає певні деталі (схоже на стиснення даних з втратами), але досягає спрощення. Він представляє безліч об'єктів даних декількома кластерами, і, отже, моделює дані за кластерами. Моделювання даних ставить кластеризацію в історичну перспективу, що базується на математиці, статистиці та чисельному аналізі.

## 2.4 Попередня обробка даних

Перш ніж розпочати роботу над завданням, потрібно переглянути набір даних. Кожен раз, коли дані збираються з різних джерел, вони збираються у сирому форматі, такий формат ускладнює або навіть унеможливлює аналіз. Більшість наборів даних спочатку обробляють, щоб на кінцевому етапі їх можна було використати у алгоритмі.

Один із способів, які часто використовуються при обробці — це нормалізація значень, яка відноситься до процесу, що робить дані більш нормальним або регулярним. Якщо є чимало нерелевантних і зайвих даних або шумних і ненадійних даних, то виявлення закономірностей під час фази навчання стає неможливим. Етапи підготовки та фільтрації даних можуть зайняти значний час обробки (рисунок 2.3). Попередня обробка даних включає в себе очищення, вибір екземпляра, нормалізацію, трансформацію, вилучення та вибір функцій тощо. Продукт попередньої обробки даних є остаточним набором тренувань.

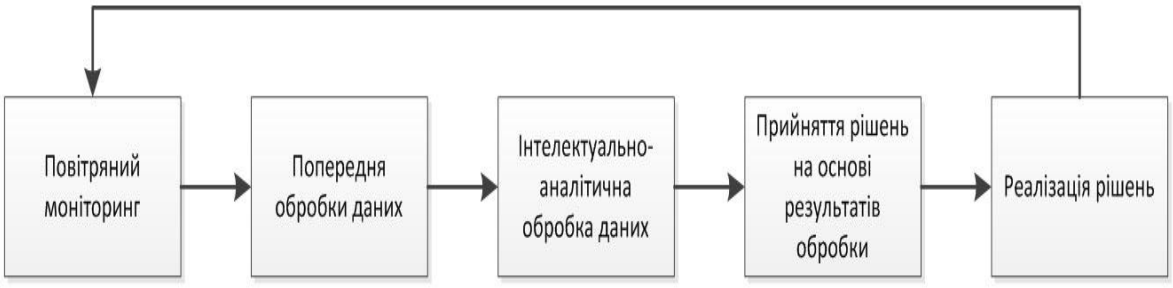



Рисунок 2.3 – Принцип роботи Інтелектуально-аналітичної системи для обробки даних

Кожен набір даних відрізняється і створює унікальні проблеми. Він може містити неформатовані дані реального світу, які можуть складатися з:

Неточні дані (відсутні дані) — є багато причин для відсутності даних, таких як дані, які не збираються постійно, помилка при введенні даних, технічні проблеми з біометрикою та багато іншого.

Наявність зашумлених даних (помилкові дані і викиди) — причини існування зашумлених даних можуть бути викликані технологічною проблемою гаджета, який збирає дані, людської помилки під час введення даних і багато іншого.

Непослідовні дані — наявність невідповідностей пояснюється такими причинами, як дублювання даних, помилкові значення, тобто порушення обмежень даних та багато іншого.

Саме тому для обробки початкових даних виконується попередня обробка даних. В деяких випадках, підготовка даних займає більше половини всього аналітичного процесу.

1) Ієрархічна кластеризація створює ієрархію кластерів або, іншими словами, дерево кластерів, також відомий як дендрограма. Кожен вузол кластера містить дочірні кластери; кластери братів розділяють точки, охоплені їхнім спільним батьком. Такий підхід дозволяє досліджувати дані на різних рівнях деталізації. Ієрархічні методи кластеризації поділяються на агломераційні (знизу вгору) і розділяють (зверху вниз). Агломераційна кластеризація починається

з одноточкових (одиночних) кластерів і рекурсивно об'єднує два або більше найбільш відповідних кластерів. Кластеризація, що розділяє, починається з одного кластера всіх точок даних і рекурсивно розбиває найбільш відповідний кластер. Процес триває до тих пір, поки не буде досягнуто критерію зупинки (часто, запитаного числа k кластерів).

2) Алгоритми розбиття даних, які розділяють дані на декілька підмножин. Оскільки перевірка всіх можливих систем підмножин є обчислювально неможливими, деякі жадібні евристики використовуються у вигляді ітеративної оптимізації. Зокрема, це означає різні схеми переміщення, які ітеративно перепризначають точки між кластерами k. На відміну від традиційних ієрархічних методів, в яких кластери не повторюються після побудови, алгоритми переселення поступово покращують кластери. Завдяки відповідним даним, це призводить до високоякісних результатів. Один з підходів до розбиття даних полягає в тому, щоб взяти концептуальну точку зору, яка ідентифікує кластер з певною моделлю, чиї невідомі параметри повинні бути знайдені.

Більш конкретно, ймовірнісні моделі припускають, що дані походять з суміші кількох популяцій, чиї розподіли та попередники ми хочемо знайти. Однією з переваг імовірнісних методів є інтерпретація побудованих кластерів. Маючи коротке представлення кластера, це також дозволяє недороге обчислення його внутрішньокластерних мір, що призводить до глобальної цільової функції.

### 2.4.1 Обробка відсутніх значень

Однією з найпоширеніших проблем є обробка відсутніх значень. По-перше, немає найкращого способу позбавитися відсутніх даних. Тому потрібно добре проаналізувати вибірку, та спробувати використати найбільш популярні методи і намагатися знайти структурне рішення.

Перш ніж перейти до методів імпутації даних, треба зрозуміти, чому дані відсутні.

Випадкова відсутність – означає, що вірогідність відсутності спостереження даних не пов'язана з відсутніми даними, але вона пов'язана з деякими спостережуваними даними

Повна випадкова відсутність — вірогідність того, що певна величина відсутня, не має нічого спільного з його гіпотетичним значенням та значеннями інших змінних.

Невипадкова відсутність — дві можливі причини полягають у тому, що відсутнє значення залежить від гіпотетичного значення (наприклад, люди з високими зарплатами, як правило, не хочуть розкривати свої доходи під час обстежень) або відсутнє значення залежить від вартості іншої змінної (наприклад, припустимо, що жінки, як правило, не хочуть розкривати свій вік)

У перших двох випадках можна безпечно видалити дані з відсутніми значеннями в залежності від їх появи, а в третьому випадку видалення спостережень з відсутніми значеннями може призвести до зміщення моделі. Тому потрібно бути дуже обережними, перш ніж знімати спостереження. Проте, імпутація не обов'язково дає кращі результати.

1) Видалення відсутніх значень стирає всі дані спостереження, котрі мають одне або більше відсутнє значення. Зокрема, якщо відсутні дані обмежені невеликою кількістю спостережень, можна просто відмовитися від аналізу цих випадків. Однак у більшості випадків часто невигідним є використання видалення. Це пояснюється тим, що припущення про повну випадкову відсутність зазвичай рідко

підтримуються. В результаті, методи видалення списків призводять до зміщення параметрів і оцінок.

2) Попарне видалення аналізує всі випадки, в яких знаходяться змінні що представляють інтерес. Таким чином максимізує всі доступні дані на основі аналізу. Силою цього методу є те, що він збільшує потужність у аналізі. Проте, він припускає, що відсутні дані є повністю випадково відсутніми. Якщо використовувати попарне видалення, то буде отримана різна кількість спостережень, що впливають на різні частини моделі, таким чином ускладнюючи інтерпретацію.
3) Видалення змінних – іноді можна скинути змінні, якщо дані відсутні для більш ніж у половини спостережень, але тільки якщо ця змінна незначна. Сказавши це, імпутація завжди є кращим вибором ніж видалення змінних.
4) Методи часових рядів – останнє спостереження і наступне спостереження це типовий статистичний підхід до аналізу даних поздовжніх повторних вимірів, коли деякі спостереження можуть бути відсутні. Поздовжні дані відстежують одну і ту ж вибірку в різні моменти часу. Обидва ці методи можуть внести зміщення в аналіз і погано працювати, коли дані мають помітну тенденцію.
5) Лінійна інтерполяція – цей метод добре працює для часових рядів з деякою тенденцією, але не підходить для сезонних даних
6) Середнє значення, медіана і мода – обчислення загального середнього значення, медіани або моди є одними із основних методів обробки пропущених значень, які не використовують характеристики часового ряду та взаємозв'язок між змінними. Обчислення виконуються дуже швидко, проте мають явні недоліки. Одним з недоліків є те, що середня імпутація зменшує дисперсію в наборі даних.

7) Лінійна регресія – для початку декілька предикторів змінної з відсутніми значеннями ідентифікуються за допомогою кореляційної матриці. Найкращі предиктори вибираються і використовуються як незалежні змінні в рівнянні регресії. У якості залежної змінної використовується змінна з відсутніми даними. Для генерації рівняння регресії використовуються випадки з повними даними для змінних предиктора; потім рівняння використовується для прогнозування відсутніх значень для неповних випадків.

У ітераційному процесі вводяться значення для відсутньої змінної, а потім всі випадки використовуються для прогнозування залежної змінної. Ці кроки повторюються до тих пір, поки не буде мало різниці між прогнозованими значеннями від одного кроку до іншого, тобто вони сходяться.

Теоретично., метод дає хороші оцінки для відсутніх значень. Переваг у методу багато, проте, як правило, недоліки переважають. По-перше, через те, що замінені значення були передбачені з інших змінних, вони схильні збігатися «занадто добре» і, таким чином, стандартна помилка дефлірована. Треба також припустити, що існує лінійна залежність між змінними, що використовуються в рівнянні регресії, коли такого бути немає.

### 2.4.2 Обробка екстремальних значень

Екстремальні значення – це спостереження, які дуже сильно відрізняються від інших даних, вони можуть вказувати на мінливість у вимірюванні, експериментальні помилки або новизну. Іншими словами, відхилення - це спостереження, яке відхиляється від загальної картини на вибірці.

Викиди можуть бути двох видів: одновимірні і багатовимірні. Одновимірні викиди можна знайти, якщо дивитися на розподіл значень в одному просторі ознак. Багатовимірні викиди можна знайти в n-мірному просторі (n-особливостей). Дивлячись на розподіли в n-мірних просторах, це може бути дуже важким для людського мозку, тому для цього використовуються моделі.

Викиди можуть також можуть бути різного виду, залежно від середовища спостереження: точкові відхилення, контекстуальні викидів або колективні відхилення. Для обробки таких значень використовуються:

1) Z-оцінка або стандартна оцінка спостереження – це метрика, яка вказує, на скільки стандартних відхилень спостереження знаходиться від середнього значення вибірки, за гаусовим розподілом. Дуже часто спостереження не описуються гаусовим розподілом, ця проблема може бути вирішена шляхом застосування перетворення даних, тобто: масштабування.
2) Dbscan (просторова кластеризація додатків на основі щільності) – у машинному навчанні та аналітиці даних методи кластеризації є корисними інструментами, які допомагають нам краще уявляти та розуміти дані. Відносини між особливостями, тенденціями та популяціями в наборі даних можуть бути графічно представлені методами кластеризації, такими як dbscan, і можуть також застосовуватися для виявлення викидів у непараметричних розподілах у багатьох вимірах. Dbscan є алгоритмом кластеризації на основі щільності, він орієнтований на пошук сусідів за щільністю на «n-мірній сфері» з радіусом ε . Кластер може бути визначений як максимальний набір "точок щільності, зв'язаних" у просторі ознак.
3) Forest Isolation – є ефективним методом для виявлення викидів або нових даних. Це відносно новий метод, заснований на бінарних деревах рішень. Реалізація у scikit learn відносно проста і зрозуміла. Основний принцип Forest Isolation полягає в тому, що викидів мало, і

вони далекі від інших спостережень. Щоб побудувати дерево, алгоритм випадковим чином вибирає функцію з простору ознак і випадкове значення розбиття, що варіюється від максимумів до мінімумів. Це робиться для всіх спостережень у навчальному наборі. Для побудови лісу створюється ансамбль дерев, що усереднює всі дерева в лісі. Потім для прогнозування він порівнює спостереження з цим значенням розщеплення в «вузлі», цей вузол буде мати двох вузлів дітей, на яких буде зроблено інше випадкове порівняння. Кількість "розщеплень", зроблених алгоритмом для екземпляра, називається: "довжина шляху". Як і очікувалося, викиди будуть мати меншу довжину, ніж інші спостереження.

## Висновки до розділу 2

У другому розділі розглянуто існуючі методи лінійної регресії, котрі, як правило, простіша у використанні і включає в себе деякі корисні статистики, які не можуть бути відображені нелінійнійною регресією, такі як p-значення для коефіцієнтів і R-квадрат

парсингу та обробки даних, моделювання та прогнозування, а саме лінійна регресія, модель авторегресії, модель ковзного середнього, REST API та інші.

Розглянуто інтерфейс Rest API – котрий вимагає використання гіпертекста, який дуже добре масштабується, оскільки клієнт і сервер дуже вільно пов'язані. За допомогою REST сервер може вільно змінювати доступні ресурси за власним бажанням. Не існує фіксованого API вище і поза тим, що визначає REST. Клієнтові потрібно знати лише початковий URI, а потім вибирати з наданих сервером варіантів навігації або виконання дій. Сервер може завантажувати код клієнту, який допомагає в навігації та відображені.

Також були розібрані та визначені основні методи обробки даних. Обробка даних є невід'ємним кроком для формування уявлень. Це один з найбільш трудомістких і важливих процесів у видобутку даних. Простими словами, підготовка даних є методом збору, очищення, обробки та консолідації даних для використання в аналізі. Він збагачує дані, перетворює їх і підвищує точність результату.

# РОЗДІЛ 3 ІНФОРМАЦІЙНО-АНАЛІТИЧНА СИСТЕМА АНАЛІЗУ МОДЕЛЮВАННЯ І ВИЯВЛЕННЯ КОРУПЦІЇ

В даному розділі наведений опис системи розробленої в рамках бакалаврської дипломної роботи, описано архітектуру, технічні вимоги до системи, показані результати її роботи. Також розповідається про функції мови програмування SAS Base, що була використана при розробці системного блоку, що відповідає за дисперсійний аналіз.

Розроблена система призначена для аналізу даних по злочинам (часових рядів), створення математичних моделей на введених даних і прогнозування на визначений строк.

В системі було повністю розроблено та впроваджено програмний продукт, простий у використанні та інтуїтивно зрозумілий. Даний продукт складає карту ризику, щодо конкретного декларанта. Окрім того, є можливість виконати прогнозування

Для вивантаження даних використовується мови програмування: python, JavaScript. Для побудови математичних моделей використовується мова програмування Python.

## 3.1 Опис реалізації

Для розміщення декларації, декларанту потрібно зайти в особистий кабінет за допомогою кваліфікованого електронного підпису. Дані розміщуються у відкритому доступі на сайті НАЗК. За допомогою веб-краулера написаного на мові програмування Python [15]. Дані завантажуються як файли з розширенням .json і за допомогою JavaScript об’єднуються в таблицю, після чого завантажуються до бази даних Oracle.

До бази даних Oracle, SAS EG підключається за допомогою спеціального конектора SAS/ACCESS, який дозволяє працювати напряму з даними. За допомогою вбудованої функції Prompt, можна побудувати інтерфейс, який буде зрозумілий співробітнику будь-якого рівня досвіду.

Для реалізації правил, використовувались такі розділи: «об'єкти нерухомості», «цінне рухоме майно - транспортні засоби», «цінні папери», «доходи, у тому числі подарунки» та «грошові активи». Одне з правил, наприклад, звучить як: «У скільки разів доходи декларанта, перевищують грошові активи?». Схожих правил було реалізовано більше 10. Якщо правило виконується, йому присвоюється певний коефіцієнт ризику. Для кінцевої таблиці, утворюється загальний коефіцієнт ризику, який являється сумою усіх інших коефіцієнтів. За загальним коефіцієнтом утворюється група ризику, яку слід перевірити у першу чергу.

## 3.2 Архітектура

На рисунку 3.1 наведена структура розробленого програмного продукту

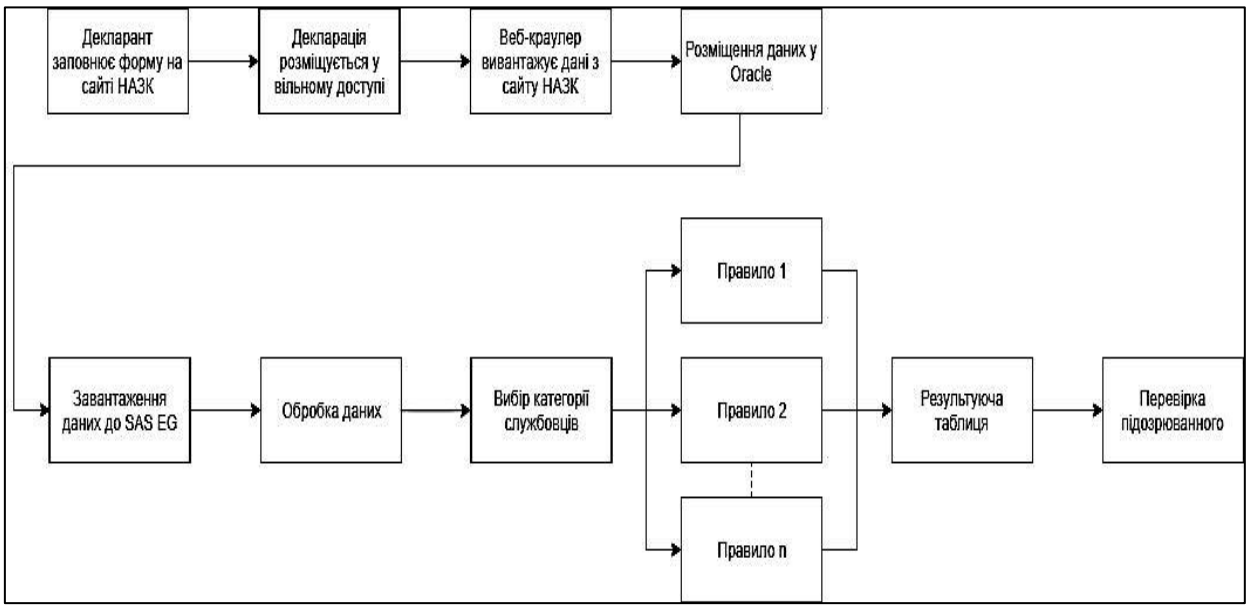


Рисунок 3.1 – Структура СППР для аналізу декларацій

## 3.3 Технічні вимоги до системи

Для нормальної роботи системи необхідна ЕОМ з характеристиками:

- процесор  або вище ;
- оперативна пам'ять: 2048 Mb або вище;
- вільний дисковий простір: 6 Gb або вище для розміщення виконавчого файлу, вхідних даних і результатів роботи;
- операційна система: Windows 7/8/10;
- клавіатура та миша;
- монітор з розподільчою здатністю 1280× 1096 пікселів і вище;
- пристрій для забезпечення безперебійного живлення ЕОМ для можливості автономної роботи у форс-мажорних обставинах;
- пристрій для запису даних на зовнішні носії інформації.

## 3.4 Інструкція по експлуатації написаного програмного продукту

Для кореляційного аналізу даних на базі ПЗ SAS Enterprise Guide та мови програмування SAS Base використовуються вбудовані функції. Проект є простим у використанні для користувачів будь-якого рівня підготовки.

Для того, щоб запустити скорингову карту потрібно відкрити SAS Enterprise Guide. Після запуску одразу відкриється вікно з порожнім проектом (рисунок 3.2).

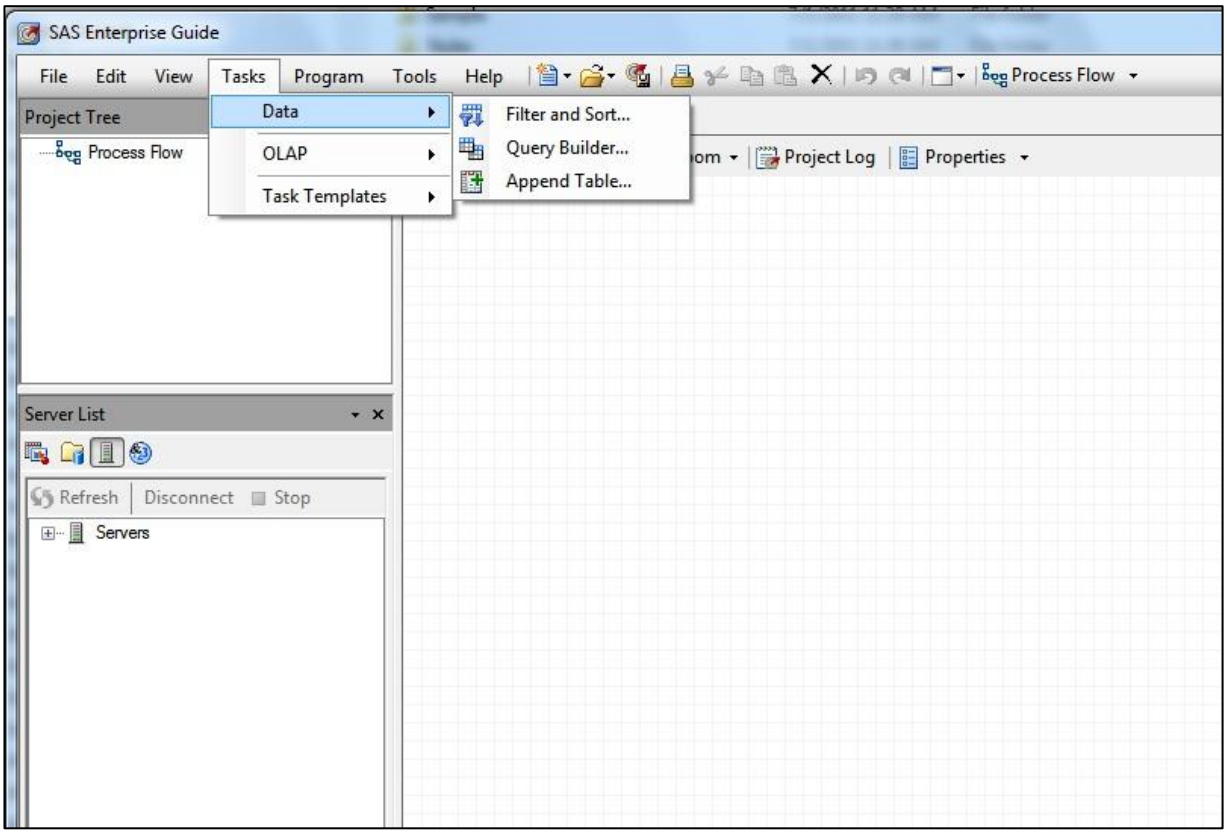



Рисунок 3.2 – Порожній проект

SAS Enterprise Guide автоматично встановлює зв'язки між більшістю елементів у вашому проекті. Коли ви дивитеся на схему процесу, ви можете легко побачити, які завдання використовують, які таблиці даних, і які результати є результатом яких завдань. Однак іноді ви можете додати свої власні посилання. Наприклад, SAS Enterprise Guide не додає посилання між таблицями даних і програмами SAS, які їх використовують. У цих випадках можна додавати посилання вручну для показу відносин.

У цьому проекті було додано посилання між таблицею даних Олімпіади та програмою SAS. Зверніть увагу, що вручну додані посилання використовують пунктирну лінію, а автоматичні посилання використовують суцільну лінію.

### 3.4.1 Завантаження даних

Першим кроком необхідно завантажити в системі SAS Enterprise Guide дані, що знаходяться на жорсткому диску ПК у форматі .txt.

Для завантаження даних потрібно перейти в головне меню: File (Файл) –> Open Data file (Відкрити файл даних). Після чого відкриється вікно вибору директорії з файлами, в якому слід обрати формат необхідного файлу з даними, та безпосередньо сам файл даних.

Програмне забезпечення автоматично згенерує код завантаження даних. Для цього використовується вбудована функція Import Wizard, котра дозволяє завантажувати дані будь-якого формату. Для початку, ми повині вибрати формат файлу імпорту (рисунок 3.3).

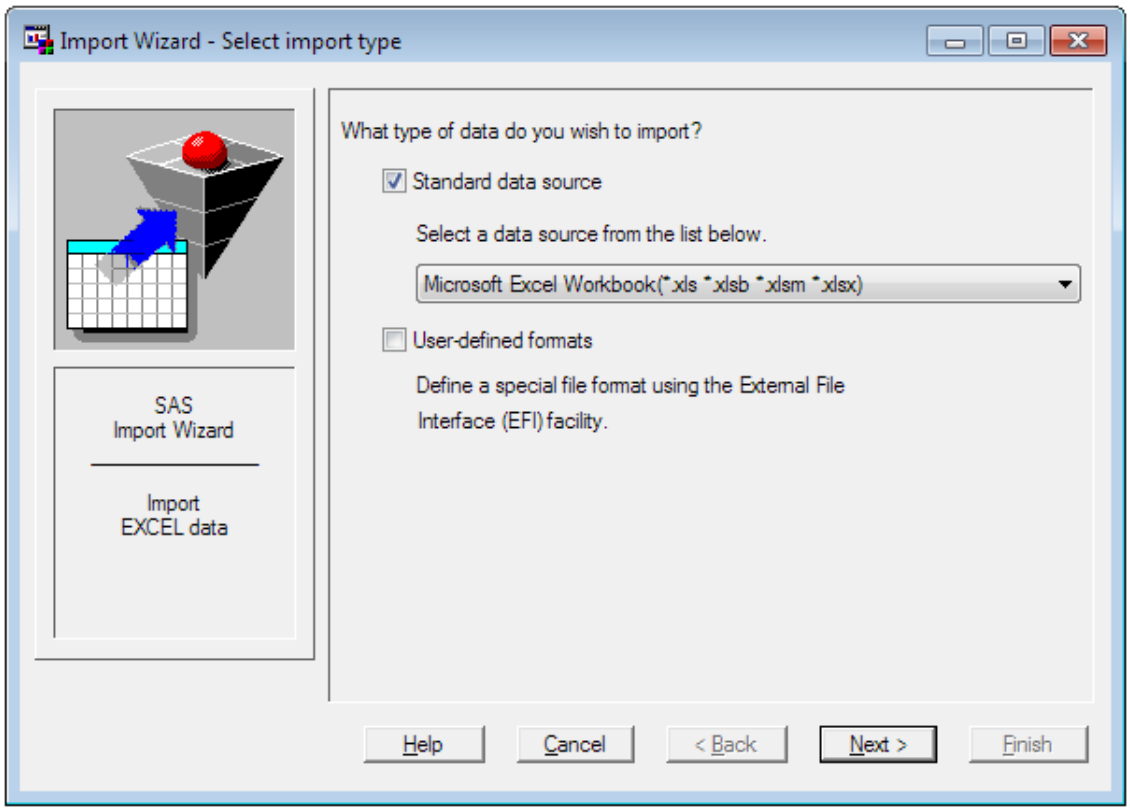



Рисунок 3.3 – Вибір формату даних

Далі визначити межі стовпців, додавши межі лінії просто натискаючи і перетягуючи їх. Якщо обрати варіант "Виправлені стовпці" як формат

вхідного тексту, буде видно лінійку макета, яка виглядає як зображено на рисунку 3.4.

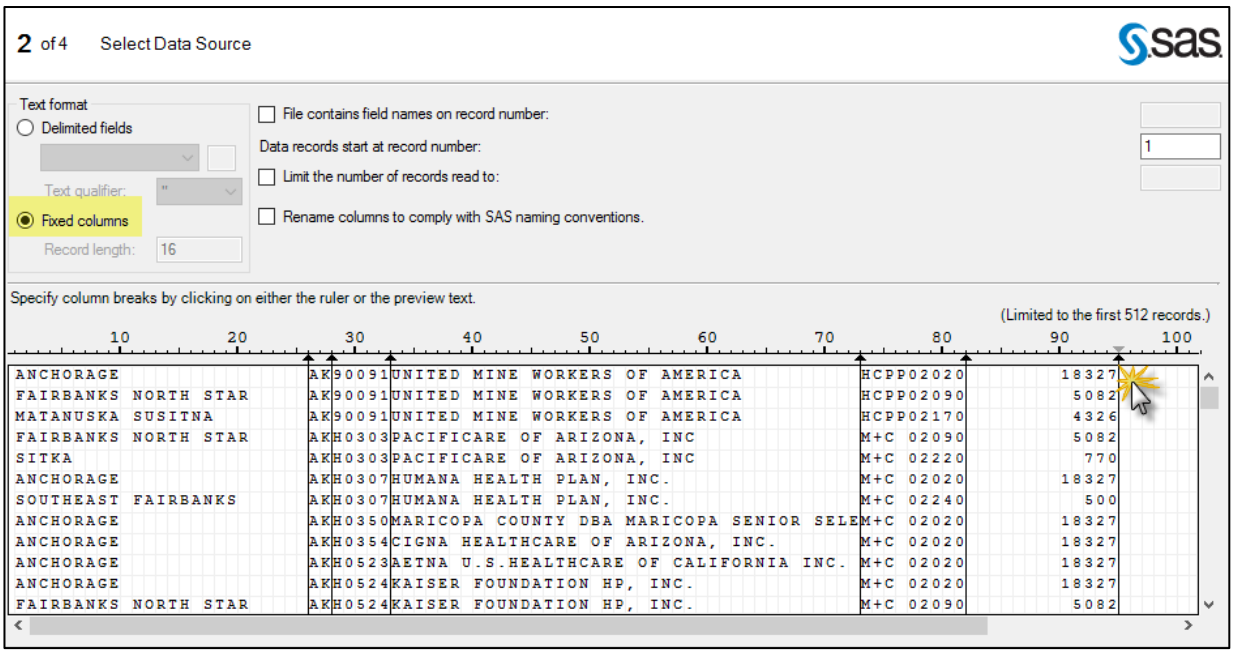
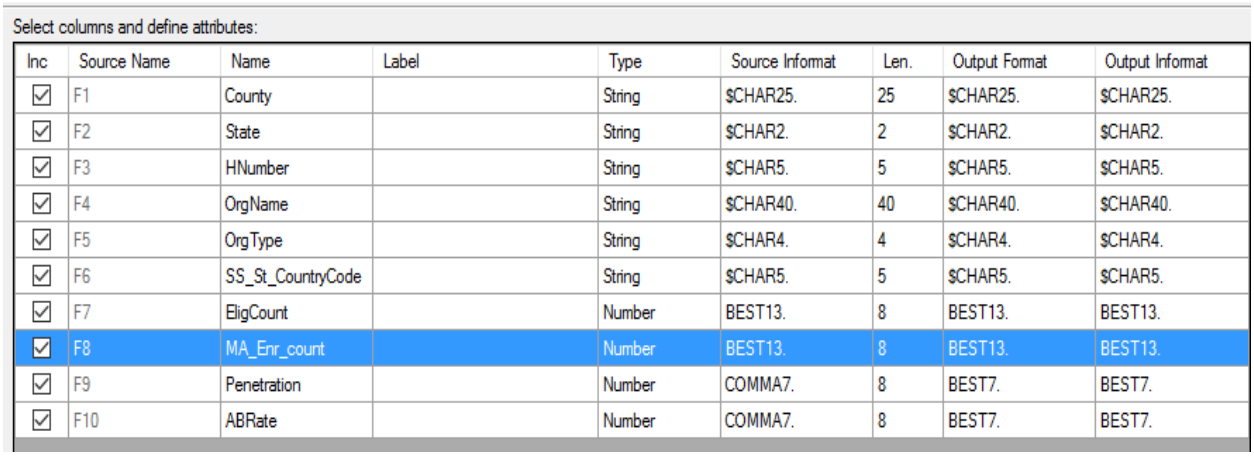



Рисунок 3.4 – Лінійка виправлення стовпців

Натиснути на межі стовпців (посилаючись на оригінальну специфікацію) і перетягнувши рядки правил, можна визначити межі стовпців. Далі відображаються відомості про імена та типи стовпців даних (рисунок 3.5).




| Inc | Source Name | Name | Label | Type | Source Informat | Len. | Output Format | Output Informat |
|---|---|---|---|---|---|---|---|---|
| ☑ | F1 | County | | String | $CHAR25. | 25 | $CHAR25. | $CHAR25. |
| ☑ | F2 | State | | String | $CHAR2. | 2 | $CHAR2. | $CHAR2. |
| ☑ | F3 | HNumber | | String | $CHAR5. | 5 | $CHAR5. | $CHAR5. |
| ☑ | F4 | OrgName | | String | $CHAR40. | 40 | $CHAR40. | $CHAR40. |
| ☑ | F5 | OrgType | | String | $CHAR4. | 4 | $CHAR4. | $CHAR4. |
| ☑ | F6 | SS_St_CountryCode | | String | $CHAR5. | 5 | $CHAR5. | $CHAR5. |
| ☑ | F7 | EligCount | | Number | BEST13. | 8 | BEST13. | BEST13. |
| ☑ | F8 | MA_Enr_count | | Number | BEST13. | 8 | BEST13. | BEST13. |
| ☑ | F9 | Penetration | | Number | COMMA7. | 8 | BEST7. | BEST7. |
| ☑ | F10 | ABRate | | Number | COMMA7. | 8 | BEST7. | BEST7. |

Рисунок 3.5 – Відомості про поля даних

Який потім повідомляє завдання імпорту даних, як генерувати відповідні оператори INPUT. Після натиску кнопки "Готово", генерується набір даних, готовий для вирішення поставлених задач (рисунок 3.6).

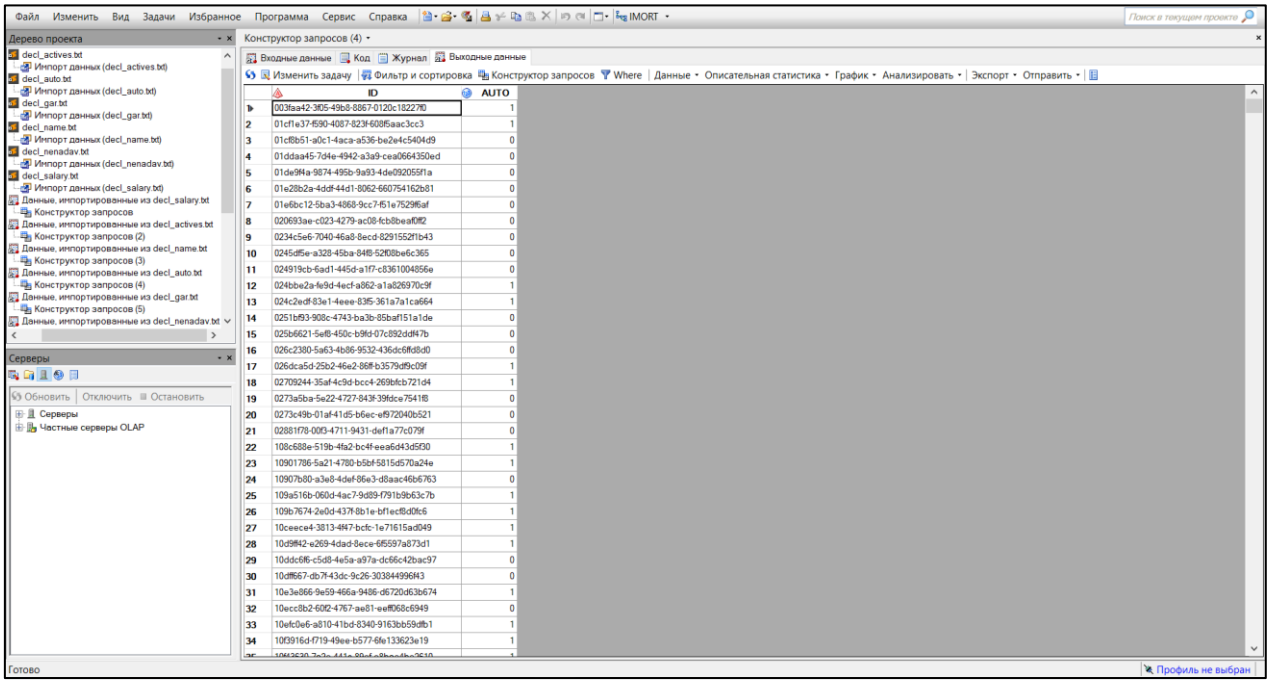


Рисунок 3.6 – Вихідна таблиця завантажених даних

Для завантаження даних з бази даних, такої як ORACLE використовується команда LIBNAME. Для цього потрібно ввести логін, пароль, шлях та назву схеми проекту БД (рисунок 3.7).

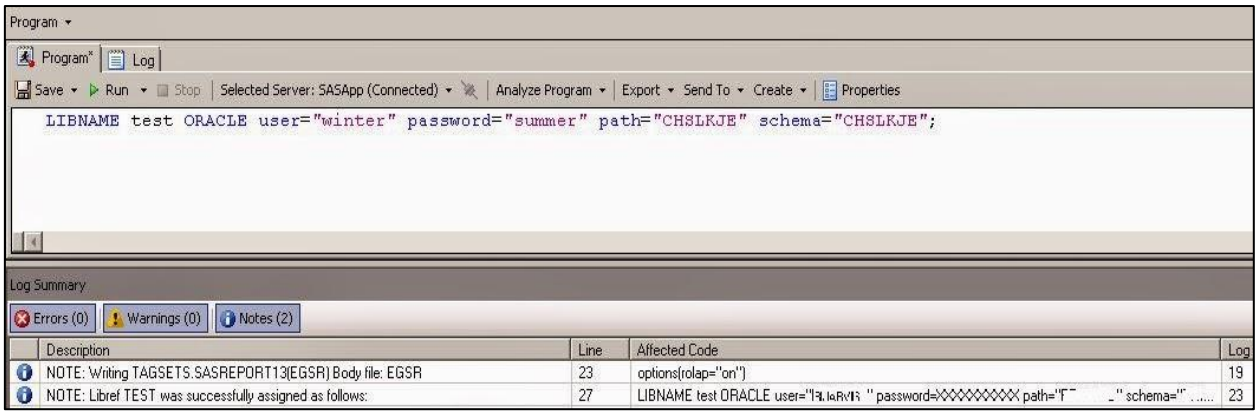



Рисунок 3.7 – Підключення до бази даних Oracle

Після підключення до бази даних на вкладці сервер будуть відображатись усі доступні таблиці. Окрім БД Oracle там відображаються стандартні бібліотеки локального сервера SAS, який називається "Локальний" у списку серверів. Всі посилання на файлові системи та бібліотеки визначаються з точки зору локального комп'ютера (рисунок 3.8).

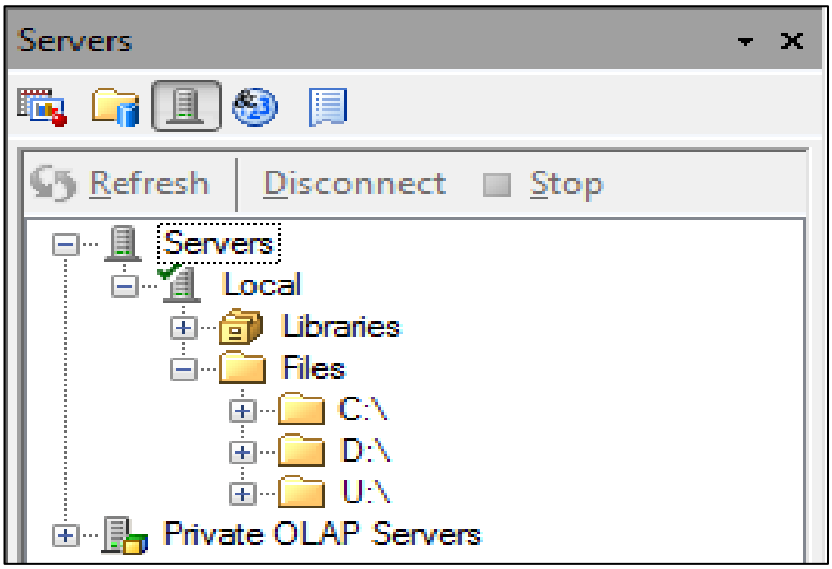



Рисунок 3.8 – Список доступних підключень на вкладці Server List

Будь-які функції SAS Enterprise Guide, які покладаються на метадані SAS, буде вимкнено. До них відносяться збережені процеси, бібліотеки на основі метаданих, інтеграція з SAS Visual Analytics тощо.

### 3.4.2 Класифікація декларацій за посадою та карта ризику

Для того щоб пришвидшити роботу програмного забезпечення, потрібно розкласифікувати декларантів. Найкращі зміни для цього – це посада та місце роботи. Оскільки данні поля заповнюються декларантами вручну і не мають чіткого формату – класифікування вручну може зайняти дуже великий проміжок часу.

Для вирішення цієї проблеми зручну використати text mining платформу з вбудованими методами текстової аналітики таку, як наприклад SAS Enterprise Guide чи Enterprise Miner.

Результати роботи програми зображені на рисунку 3.9.

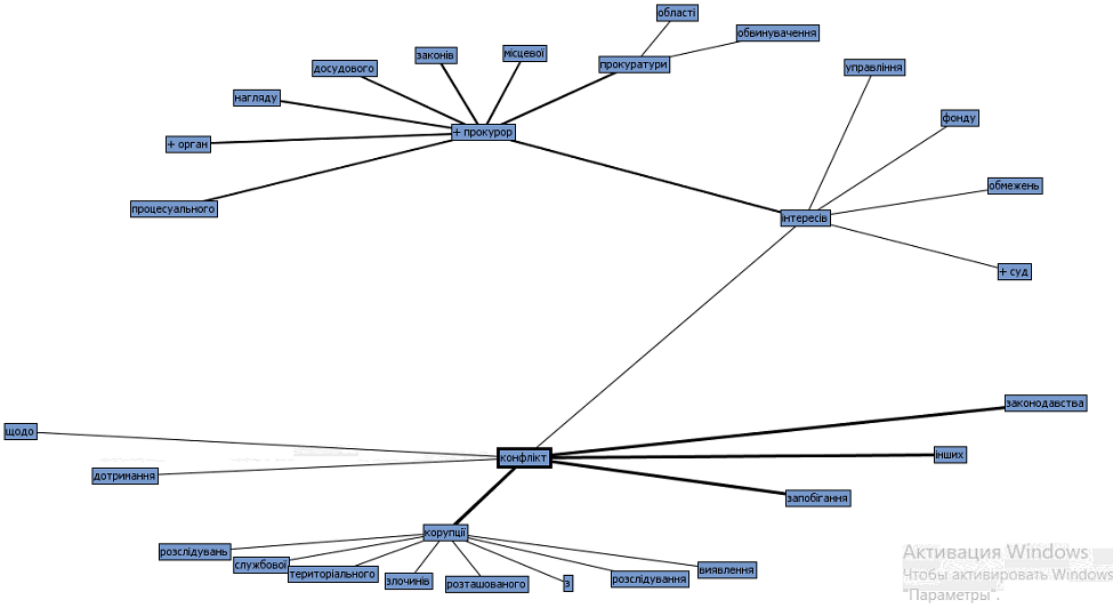



Рисунок 3.9 – Лист синонімів для пошуку посади

Після кластеризації держслужбовців ми можемо розробити нашу карту оцінки ризиків.

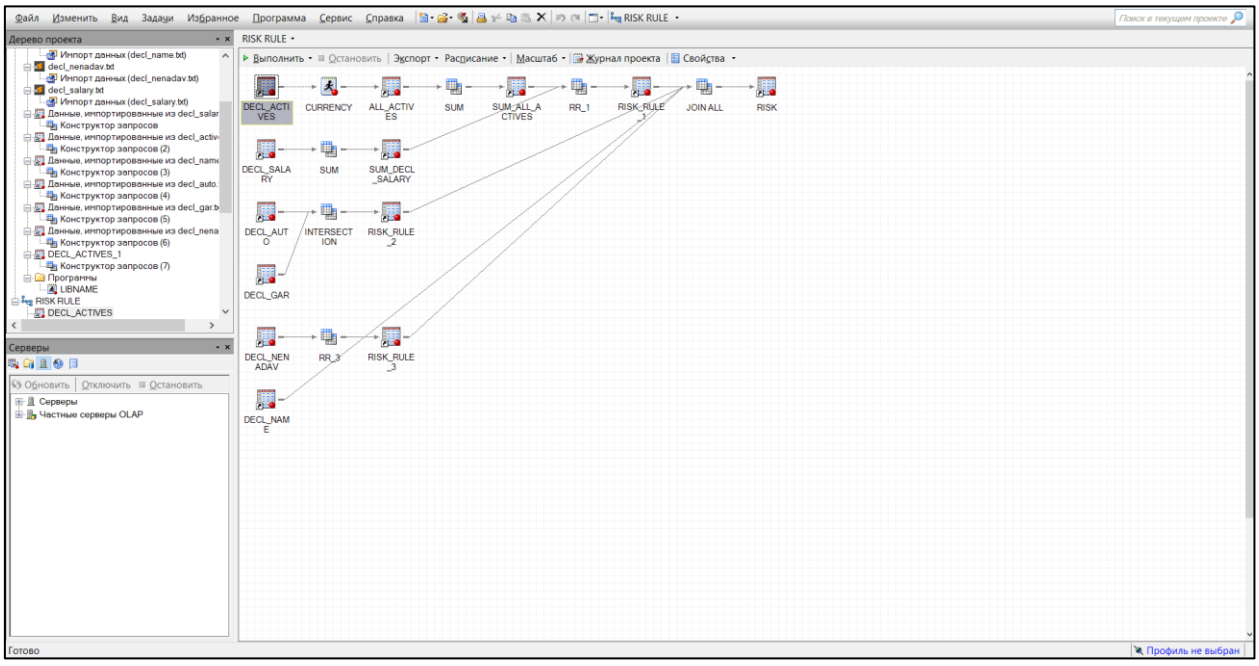


Рисунок 3.10 – Карта оцінки ризиків

Карта оцінки ризиків утворю собою кінцеву таблицю з результатом, гоотовим до опрацювання аналітиком

### 3.4.3 Побудова прогнозуючої моделі

Дослідження зосереджувалося на різних змінних, таких як ВВП або валовий внутрішній продукт на душу населення, рівень інфляції та рівень безробіття разом із залежною змінною Індексу сприйняття корупції.

Набір даних, який був використаний для цього дослідження, був отриманий з різних статистичних наборів даних саме в Index mundi; які отримали свої дані з CIA World Factbook та Transparency International; який містив виміряний рейтинг корупції в країні (таблиця 3.1).

Таблиця 3.1 – Структура регресійної моделі рівня корупції

| Змінна | Позначення | Означення | Вимірювання |
|---|---|---|---|
| Залежна змінна / регресор | | | |
| Індекс Сприйняття корупції | CPI | Вимірює, наскільки поширена корупція в державному секторі даної країни. Країни оцінені від 0 до 10, де 10 – найменш корумпована країна. | Рейтинг |
| Незалежна змінна / предиктор | | | |

| | | | |
|---|---|---|---|
| ВВП | GDPpc | Вартість товарів і послуг, вироблених в одній країні поділений на населення даної країни. | Долари США |
| Рівень безробіття | UnempRate | Процент безробітних серед робочої сили. | Процент |
| Рівень інфляції | InflaRate | Темпи зростання споживчих цін. | Процент |

Використовуючи змінні, наведені вище в таблиці 3.1, можна скласти економетричну модель, яка визначатиме мету статті, яка полягає в аналізі впливу корупції, регресії та регресорів, які є ВВП на душу населення, рівень безробіття та рівень інфляції.

Таблиця 3.2 – знаки при предикторах моделі

| Зміна | Знак | Пояснення |
|---|---|---|
| ВВП | + (додатне) | Взаємозв'язок ВВП з CPI є позитивним, тому що якщо уряд буде впроваджувати належне керування і вчасно приймати міри, то корупція буде мінімальною, і доходи, які уряд зможе зібрати з населення, будуть ефективно використані для підвищення економічного виробництва країни. Тому, чим вище рейтинг країни в CPI, тим вищий її економічний випуск. |
| Рівень безробіття | – (від'ємне) | Існує негативний зв'язок між CPI і рівнем безробіття – якщо в країні існує корупція, виділення робочих місць для людей буде проблемою державного сектора. Крім того, якщо країна корумпована, то створюється вплив на інвестиції і, якщо інвесторам будуть заважати надавати капітал для нових підприємств, можливостей для безробітних буде менше. |

| | | |
|---|---|---|
| Рівень інфляції | – (від'ємне) | Рівень інфляції має непрямий зв'язок з CPI, оскільки збільшення рівня інфляції означає, що уряд може використовувати макроекономічну фіскальну політику, щоб збільшити шляхи доходів для існуючої корупційної системи. |

Індекс сприйняття корупції = $\beta_0$ + $\beta_1$ · Валовий внутрішній продукт на душу населення – $\beta_2$ - рівень безробіття - $\beta_3$ + рівень інфляції + $u_i$ (таблиця 3.2). Використовуючи мітки для змінних, рівняння можна виразити додатково з визначеними знаками.

$$\text{CPI} = \beta_0 + \beta_1 \text{GDP} - \beta_2 \text{UnempRate} - \beta_3 \text{InflaRate} + u_i$$

Отримані результати (рисунок 3.9) з початкової регресії найменших квадратів (OLS) на розрахункових коефіцієнтах кожної з незалежних змінних. Значення цих коефіцієнтів можна замінити в оціночній моделі, де:

$$\text{CPI} = 3.5339 + 0.0001\ \text{GDP} - 0.01125\ \text{UnempRate} - 0.10304\ \text{InflaRate} + u_i$$

| Source | SS | df | MS | Number of obs = 118 |
|---|---|---|---|---|
| | | | | F( 3, 114) = 84.75 |
| Model | 405.197884 | 3 | 135.065961 | Prob > F = 0.0000 |
| Residual | 181.676347 | 114 | 1.59365217 | R-squared = 0.6904 |
| | | | | Adj R-squared = 0.6823 |
| Total | 586.874231 | 117 | 5.01601907 | Root MSE = 1.2624 |

| CPI | Coef. | Std. Err. | t | P>\|t\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| GDPpc | .0000918 | 8.02e-06 | 11.46 | 0.000 | .0000759 | .0001077 |
| UnempRate | -.0112513 | .0082036 | -1.37 | 0.173 | -.0275025 | .0049999 |
| InflaRate | -.1030416 | .030841 | -3.34 | 0.001 | -.1641375 | -.0419458 |
| _cons | 3.533943 | .3461715 | 10.21 | 0.000 | 2.84818 | 4.219706 |

Fig A[1]

Рисунок 3.11 – Результати регресії

Під час тестування на значущість певної змінної ця змінна повинна відповідати певному критерію значущості, тобто p-значення поділене на два,

оскільки статистична система показує двостороннє p-значення, тоді як має бути тільки одностороннє. Це значення має бути менше альфа (α), яке дорівнює 0,05, оскільки існує 95% довірчий інтервал. З результатів видно, що існують два регресори: GDPpc і InflaRate, які виявляються значущими в моделі, оскільки кожне має значення p, яке більше 0.05. Якщо коли-небудь результат доведено до значення p, яке більше 0.05, нульова гіпотеза, де βi = 0, повинна бути прийнята так, щоб уникнути помилки типу 1.

Валовий внутрішній продукт на душу населення є однією з значущих незалежних змінних, знайдених у результатах. Вона має p-значення, яке дорівнює 0.000, це означає, що існують надзвичайно сильні докази проти нульової гіпотези, і тому це доводить значення регресора GDPpc. Крім того, GDPpc має позитивний коефіцієнт, який підтверджує його початкові очікування і показує, що існує позитивний зв'язок між CPI та ВВП. У зв'язку з цим, можна сказати, що збільшення одиниці ВВП збільшить CPI на 0.0000918 одиниць.

Інфляція, яка є іншою важливою змінною, має значення p, яке перевищує 0.05. P-значення регресора дорівнює 0.0005, і тому існують сильні докази проти нульової гіпотези та значення регресора InflaRate. Коефіцієнт InflaRate дорівнює - 0.1030416, що означає негативний вплив на значення, оскільки існує відповідний негативний зв'язок між InflaRate і CPI. З цього можна зробити висновок, що відсоткове збільшення інфляції зменшить CPI на 0,1030416 одиниць.

Рівень безробіття є змінною, яка виявилася незначною через її p-значення. P-величина становила 0.0865, що більше, ніж α = 0.05, де, згідно з раніше викладеним критерієм, нульову гіпотезу слід прийняти, де β2 = 0, тому регресор, UnempRate, незначний. Коефіцієнт цієї змінної підтвердив що існує негативний зв'язок між UnempRate і CPI, де відсоток збільшення UnempRate знижує CPI на 0.0112513 одиниць.

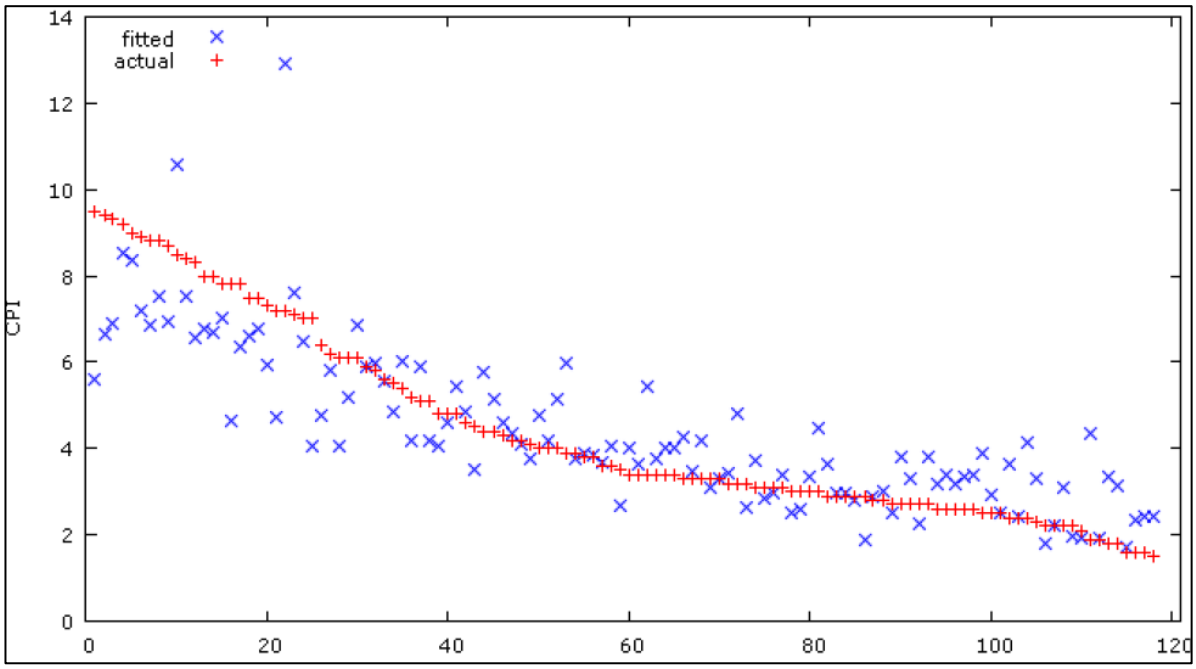



Рисунок 3.12 – Графік залежності між реальними значенням та змодельованими

Оцінені значення відображаються, і, як і залишки, вони можуть бути додані до матриці даних для подальшого аналізу ». Це показує, що лінійна регресія на CPI наближається до фактичної і є низхідною у різних країнах.

## 3.5 Оцінка прогнозуючої моделі

Лінійна регресія робить кілька припущень щодо даних, що знаходяться під рукою. Тому потрібно оцінити припущення регресії та надає вбудовані ділянки для діагностики регресії в мові програмування Python. Після виконання регресійного аналізу завжди слід перевіряти, чи добре працює модель для наявних даних.

Перший крок регресійної діагностики полягає в тому, щоб перевірити значущість регресійних бета-коефіцієнтів, а також R2, що говорить нам, наскільки добре модель лінійної регресії вписується в дані або, наприклад,

модель лінійної регресії робить припущення, що зв'язок між предикторами (x) і змінною результату є лінійною. Це може бути не так. Відносини можуть бути поліноміальними або логарифмічними. Крім того, дані можуть містити деякі впливові спостереження, такі як викиди (або крайні значення), які можуть впливати на результат регресії.

Таким чином, слід уважно діагностувати побудовану модель регресії, щоб виявити потенційні проблеми та перевірити, чи виконані припущення, зроблені моделлю лінійної регресії. Для цього ми, як правило, вивчається розподіл помилок залишків, які можуть розповісти більше про дані.

### 3.5.1 Дисперсійний аналіз

Методика перевірки на різницю в більш ніж двох незалежних засобах є розширенням процедури незалежних зразків, обговореної раніше, яка застосовується, коли існують точно дві незалежні групи порівняння. Метод ANOVA застосовується, коли існують дві або більше двох незалежних груп. Процедура ANOVA використовується для порівняння засобів груп порівняння і проводиться з використанням того самого п’яти крокового підходу, який використовується в сценаріях, обговорених у попередніх розділах. Однак, оскільки існує більше двох груп, обчислення статистичних даних тесту є більш залученими. Статистика випробування повинна враховувати розміри вибірки, засоби вибірки та стандартні відхилення зразків у кожній з груп порівняння (рисунок 3.11).

| | Sum of squares | df | Mean square |
|---|---|---|---|
| **Regression** | 405.198 | 3 | 135.066 |
| **Residual** | 181.676 | 114 | 1.59365 |
| **Total** | 586.874 | 117 | 5.01602 |
| R^2 = 405.198 / 586.874 = 0.690434 | | | |
| F(3, 114) = 135.066 / 1.59365 = 84.7525 [p-value 6.72e-029] | | | |

Рисунок 3.13 – Результати дисперсійного аналізу

Як видно з таблиці на рисунок 3.11 вище, тест ANOVA розрахував p-значення, яке дорівнює 6,72e-029. Це є частиною критерію, що p-значення має бути менше 0,05, тому з наведеного результату можна зробити висновок, що вся модель є значущою.

### 3.5.2 Припущення щодо нормальності

Тест на припущення про нормальність необхідно перевірити, оскільки це повинно показати, що регресія МНК розподілена ідентично і незалежно, і t-тест і F-тест має дійсне відповідне значення p. Для тестування припущення про нормальність нашої класичної моделі регресії в лінійному просторі. Тест показує наступну форму функції щільності ймовірності, коли існує нормальність, починаючи з хвоста над стовпчиками, які представляють фактичний розподіл, таким чином, прогнозування є лінійним згідно з фактичним значенням (рисунок 3.12).

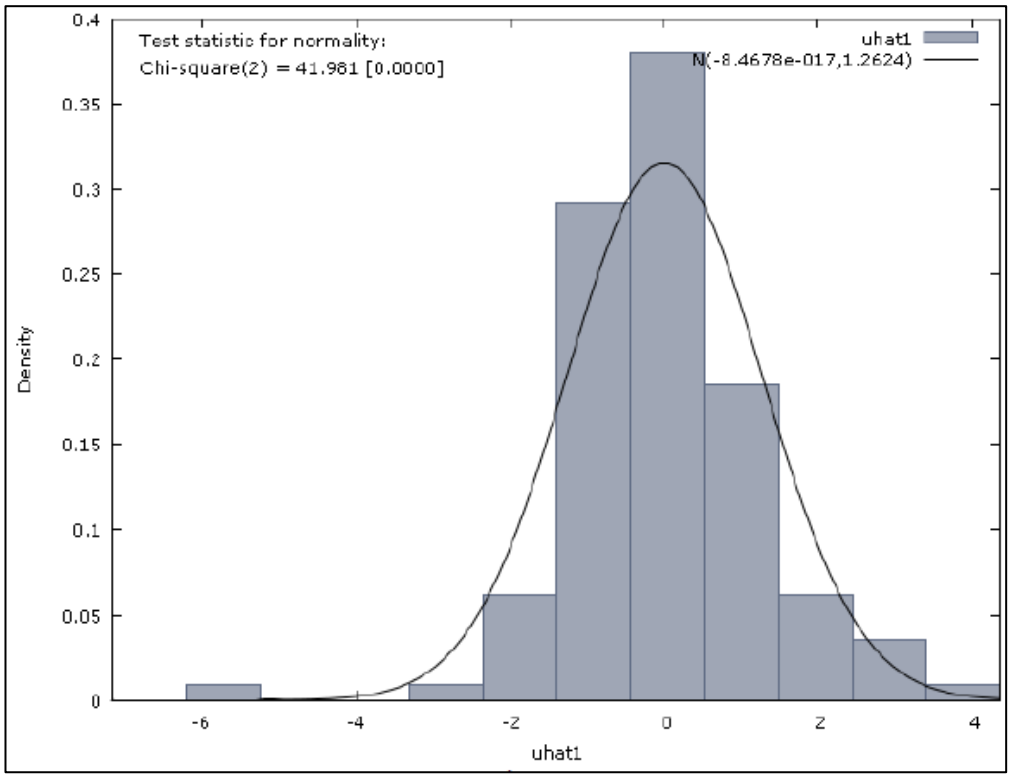



Рисунок 3.14 – Розподіл тесту на нормальність

### 3.5.3 Тест на мультиколінеарність

Тест на мультиколінеарність дотримується другого представленого критичного припущення про лінійну модель. У реальному світі мультиколінеарність завжди присутня у всіх даних. З огляду на цей факт, тест на мультиколінеарність стосується мінімізації взаємозв'язку між двома або більше змінними. Незбагненність того, що є порушення такого характеру, призведе до небезпечних наслідків, що виникнуть в парадоксі мультиколінеарності (рисунок 3.13).

| Variable | VIF | 1/VIF |
|---|---|---|
| gdp | 1.39 | 0.720316 |
| inflarate | 1.23 | 0.811620 |
| unemrate | 1.14 | 0.874511 |
| Mean VIF | 1.25 | |

Рисунок 3.15 – Тест на мультиколіеарність

Коефіцієнт інфляції містить критерій, який слід дотримуватися для того, щоб модель мала допустиму мультиколінеарність, і цей критерій говорить, що vif має бути менше 10. Дивлячись на результат, можна бачити, що немає мультиколінеарності між та серед використовуваних змінних.

### 3.5.4 Тест на гетероскедастичність

Гетероскедастичність є порушенням припущення про гомоскедастичності. Враховуючи, що дані, що використовуються в цьому регресійному аналізі, є перехресним, існує необхідність перевірити це порушення, оскільки воно переважає в даних поперечного перерізу. Причина цієї поширеності в даних поперечного перерізу обумовлена наступним:

1. Викиди
2. Заборонене змінне зміщення
3. Неправильна трансформація
4. Відхилення в поведінці вивчення помилок суб'єктів перехресних вхідних даних

Неможливість побачити гетероскедастичність у регресійній моделі призведе до того, що МНК не буде найкращою незміщеною оцінкою так, що не вистачає його властивості ефективності, оскільки стандартна помилка

оцінок буде неправильною. Це призведе до подальшого краху висновку, оскільки внаслідок цього порушення результати будуть вводити в оману.

Для того, щоб мати змогу діагностувати, чи містить модель це порушення, використовується тест Breusch-Pagan / Cook-Weisberg для гетероскедастичності, який покаже неформальний підхід визначення гетероскедастичності (рисунок 3.14).

| Breusch-Pagan / Cook-Weisberg test for heteroskedasticity |
|---|
| Ho: Constant variance |
| Variables: fitted values of cpi |
| chi2(1) = 83.92 |
| Prob> chi2 = 0.0000 |

Рисунок 3.16 – тест Breusch-Pagan / Cook-Weisberg

Отримане p-значення, яке генерується з тесту Breush-Pagan, менше 0,05, що означає, що нульову гіпотезу слід відхилити тобто в моделі є порушення гетероскедастичності, тому що відхилення нульової гіпотези означає, що гомоскедастичність також відхиляється. З іншого боку, показує, що існує закономірність в залишку і CPI, яка представляє собою гетероскедастичність (рисунок 3.15).

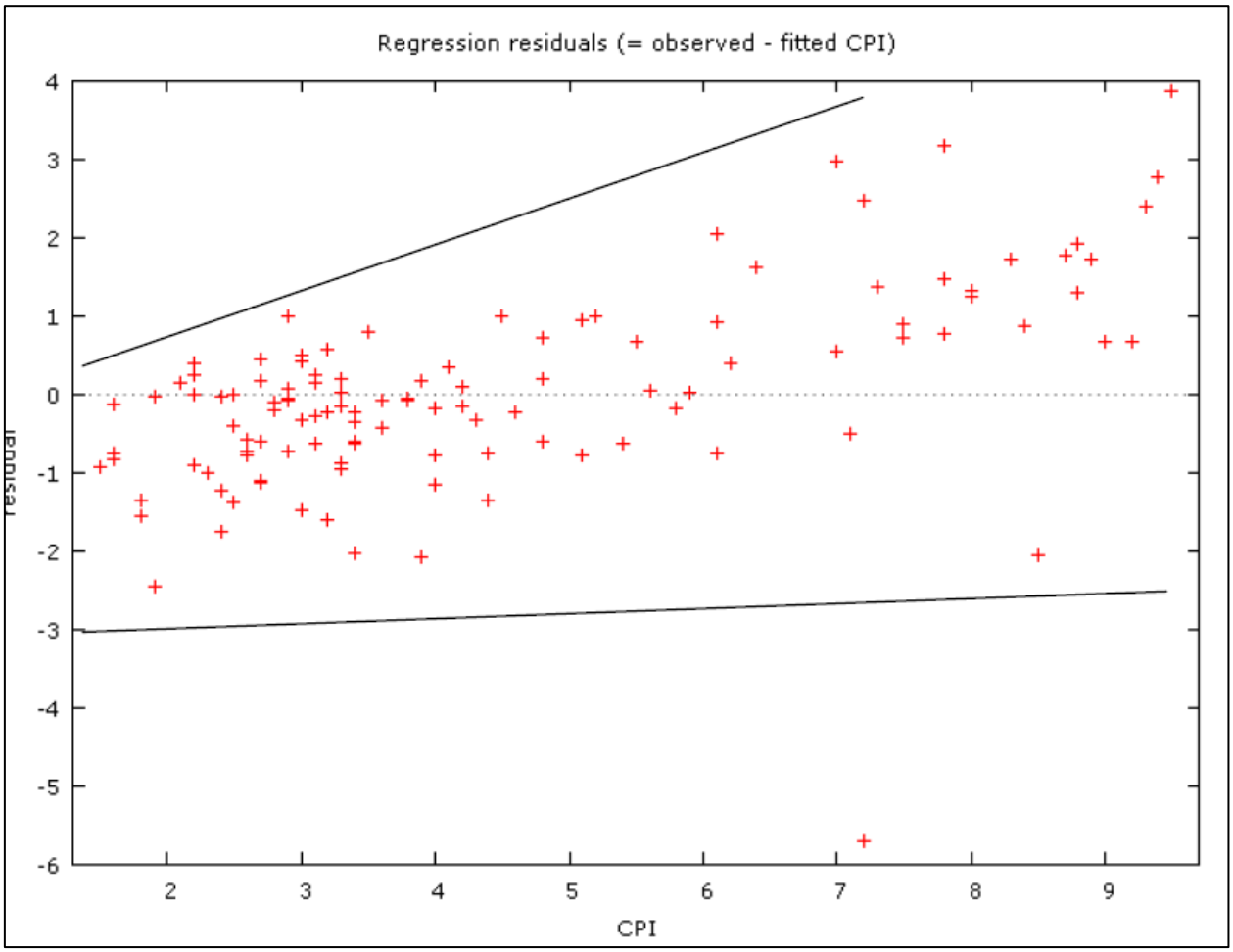



Рисунок 3.17 – Графік залишків відносно CPI

Висновки до розділу 3

У третьому розділі було описано спроектовану таблицю скорингових значень ризиків, яку розроблено в аналітичній системі SAS Enterprise Guide. Спроектована програма дозволяє проводити завантажувати дані будь-якого формату або підключитися до бази даних. За допомогою методів текстової аналітики, система дозволяє класифікувати місце роботи декларанта та рівень ризику декларації. Також, система виконує попередню обробку даних,

розраховує статистичні показники та складає кінцевий звіт зв виконаними результатами.

Розроблено модель, що дозволяє спрогнозувати Індекс сприйняття корупції за допомогою основних макроекономічних показників країни.

Описано архітектуру системи та технічні характеристики апаратного забезпечення, котре забезпечить швидке та належне виконання спроектованої системи.

Було проведено оцінку прогнозуючої моделі, реалізованої за допомогою мови програмування Python. Використано такі статистичні тести як дисперсійний аналіз, припущення на нормальність, тест на мультиколінеарність та тест на гетероскедастичність.

Розроблений програмний продукт було випробувано на реальних текстових даних Національного антикорупційного бюро України, що є одним із найвищих правоохоронних органів в сфері виявлення та розкриття корупційних злочинів.

# РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

У даному розділі проводиться оцінка основних характеристик програмного продукту, розробленого в рамках дипломної роботи. Програмний продукт написаний на мові програмування SAS у середовищі розробки SAS Enterprise Guide.

В даному розділі проводиться аналіз варіантів реалізації модулю з метою вибору оптимального, з економічної точки зору. А саме проводиться функціонально-вартісний аналіз (ФВА).

Функціонально-вартісний аналіз — це метод комплексного техніко-економічного дослідження об’єкта з метою розвитку його корисних функцій при оптимальному співвідношенні між їхньою значимістю для споживача і витратами на їхнє здійснення.

Є одним з основних методів оцінки вартості науково-дослідної роботи, оскільки ФВА враховує як технічну оцінку продукту, що розробляється, так і економічну частину розробки.

Крім того, даний метод дозволяє вибрати оптимальний, як з погляду розробника, так і з точки зору покупця варіант розв’язання будь-якої задачі, а також дозволяє оптимізувати витрати й час виконання робіт.

Мета ФВА полягає у забезпеченні правильного розподілу ресурсів, виділених на виробництво продукції або надання послуг, на прямі та непрямі витрати. У даному випадку – аналізу функцій програмного продукту й виявлення усіх витрат на реалізацію цих функцій.

Фактично цей метод працює за таким алгоритмом:

1. визначається послідовність функцій, необхідних для виробництва продукту. Спочатку – всі можливі, потім вони розподіляються по двом групам: ті, що впливають на вартість продукту і ті, що не впливають. На цьому ж етапі оптимізується сама послідовність

скороченням кроків, що не впливають на цінність і відповідно витрат.

2. для кожної функції визначаються повні річні витрати й кількість робочих часів.
3. для кожної функції на основі оцінок попереднього пункту визначається кількісна характеристика джерел витрат.
4. після того, як для кожної функції будуть визначені їх джерела витрат, проводиться кінцевий розрахунок витрат на виробництво продукту.

## 4.1 Постановка задачі техніко-економічного аналізу

У роботі застосовується метод ФВА для проведення техніко-економічний аналізу розробки.

Відповідно цьому варто обирати і систему показників якості програмного продукту.

Технічні вимоги до продукту наступні:

- програмний продукт повинен функціонувати на персональних комп’ютерах із стандартним набором компонент;
- забезпечувати високу швидкість обробки великих об’ємів даних у реальному часі;
- забезпечувати зручність і простоту взаємодії з користувачем або з розробником програмного забезпечення у випадку використовування його як модуля;
- передбачати мінімальні витрати на впровадження програмного продукту.

### 4.1.1 Обґрунтування функцій програмного продукту

Головна функція $F_0$ – розробка програмного продукту, який аналізує процес за вхідними даними та будує його модель для подальшого прогнозування. Виходячи з конкретної мети, можна виділити наступні основні функції ПП:

$F_1$ – вибір мови програмування;

$F_2$ – вибір оптимального середовища розробки;

$F_3$ – інтерфейс користувача.

Кожна з основних функцій може мати декілька варіантів реалізації.

Функція $F_1$:

а) мова програмування Python;

б) мова програмування SAS.

Функція $F_2$:

а) IDE;

б) TextReader.

Функція $F_3$:

а) віконний додаток;

б) консольний додаток.

### 4.1.2 Варіанти реалізації основних функцій

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 4.1). На основі цієї карти побудовано позитивно-негативну матрицю варіантів основних функцій (таблиця 4.1).

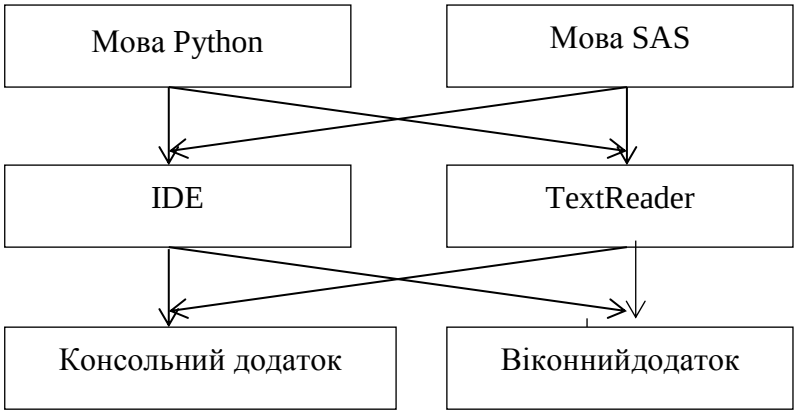



Рисунок 4.1 – Морфологічна карта

Морфологічна карта відображує всі можливі комбінації варіантів реалізації функцій, які складають повну множину варіантів ПП.

Таблиця 4.1 – Позитивно-негативна матриця

| **Основні функції** | **Варіанти реалізації** | **Переваги** | **Недоліки** |
|---|---|---|---|
| *F1* | *А* | Простота реалізації в рамках даної задачі | Складнощі в реалізації алгоритмів |
| | *Б* | Простота реалізації в рамках даної задачі | Швидкодія |
| *F2* | А | Виділення синтаксису | Довгий процес виправлення помилок |
| | Б | | Кожного разу запуск з консолі |
| *F3* | *А* | Інтерфейс інтуїтивно зрозумілий кожному | Складності в оформленні і повільність роботи |
| | *Б* | Зручність для пересічного користувача | Повільніше працює |

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому, що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція *F1:*

Оскільки в рамках даної задачі налаштування Python не займають багато часу, обираємо варіант Б.

Функція *F2:*

При роботі з даними необхідно постійно виправляти та доповнювати код, тому вибираємо варіант А.

Функція *F3:*

Оскільки, щодо інтерфейсу програмного продукту не має вишуканих вимог (ПП є виключно робочою частиною, в подальшому планується абсолютна автоматизація), то обидва варіанти А і Б влаштовують.

Таким чином, будемо розглядати такі варіанти реалізації ПП:

1. F1Б – F2A – F3A
2. F1Б – F2A – F3Б

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

## 4.2 Обґрунтування системи параметрів ПП

### 4.2.1 Опис параметрів

На підставі даних про основні функції, що повинен реалізувати програмний продукт, вимог до нього, визначаються основні параметри виробу, що будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

– *X1* – швидкодія мови програмування;

– *X2* – час обробки даних;

– *X3* – потенційний об’єм програмного коду.

*X1*: Відображає швидкодію операцій залежно від обраної мови програмування.

*X2:* Відображає час, який витрачається на дії.

*X3:* Показує розмір програмного коду який необхідно створити безпосередньо розробнику.

### 4.2.2 Кількісна оцінка параметрів

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію ПП як показано у таблиці 4.2.

Таблиця 4.2 – Основні параметри ПП

| Назва Параметра | Умовні позначення | Одиниці виміру | Значення параметра | | |
|---|---|---|---|---|---|
| | | | гірші | середні | кращі |
| Швидкодія мови програмування | X1 | Оп/мс | 19000 | 11000 | 2000 |
| Час обробки запитів | X2 | мс | 1000 | 420 | 60 |
| Потенційний об’єм прогр. коду | X3 | кількість строк коду | 350 | 200 | 100 |

За даними таблиці 4.2 будуються графічні характеристики параметрів – рис. 4.2 – рис. 4.4.

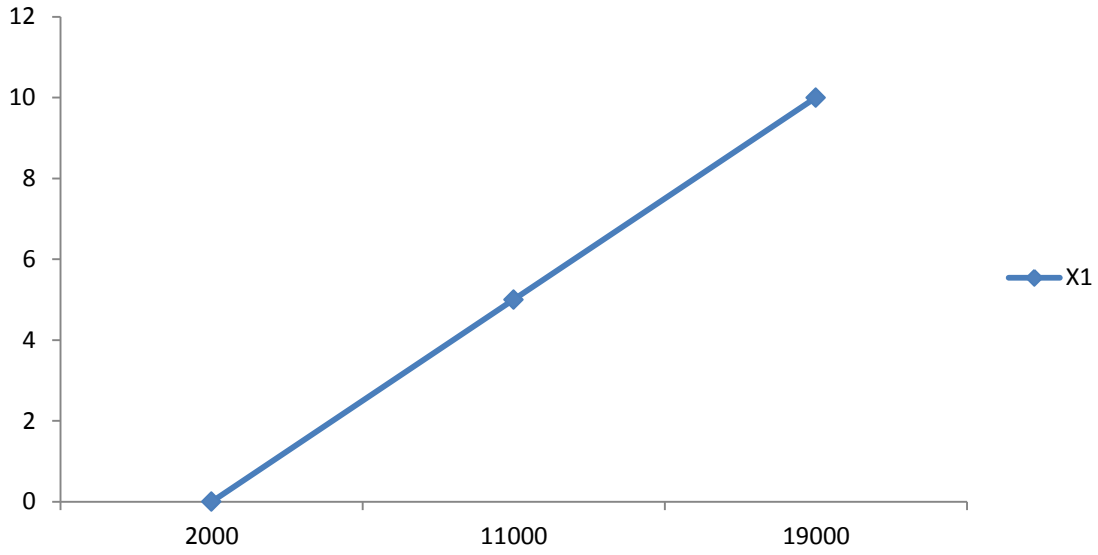



Рисунок 4.2 – X1, швидкодія мови програмування

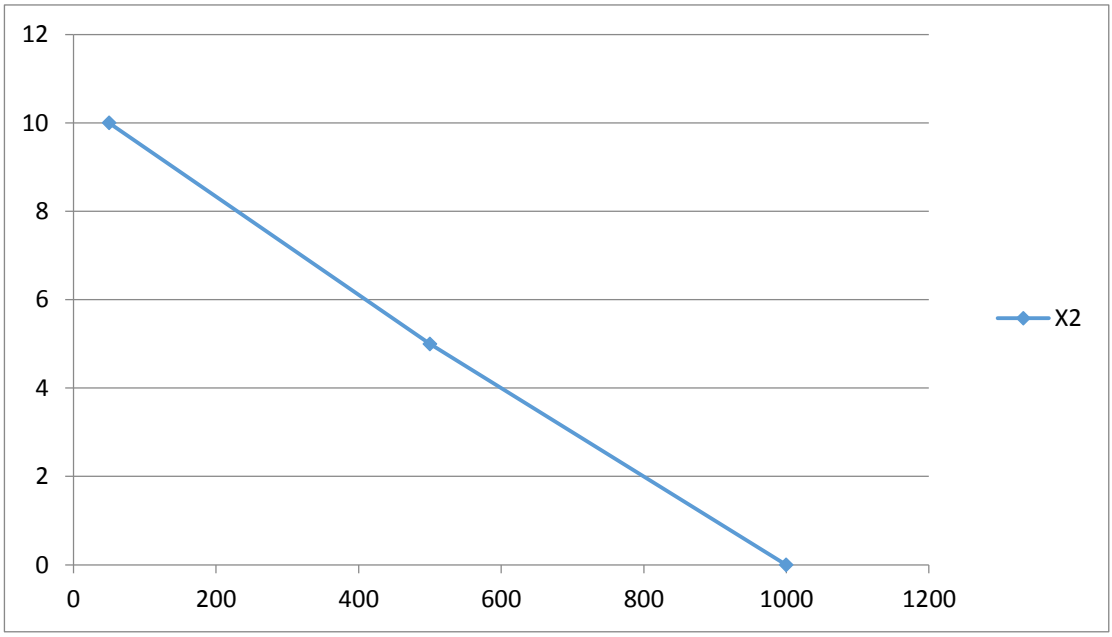



Рисунок 4.3 – X2, час виконання запитів користувача

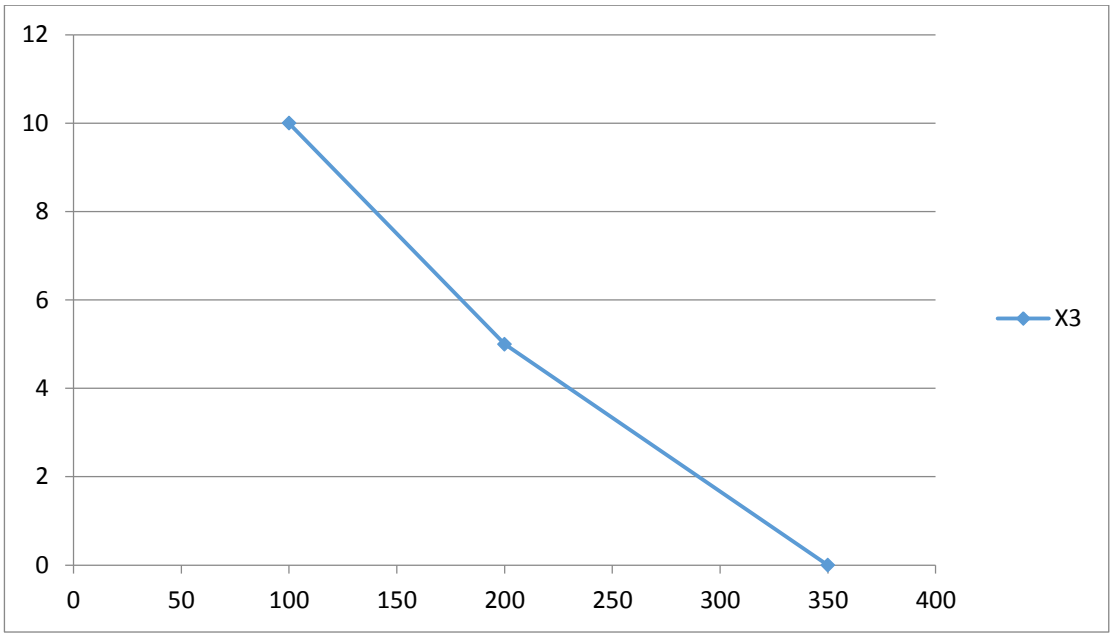



Рисунок 4.4 – X3, потенційний об’єм програмного коду

### 4.2.3 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає найбільш точні результати при знаходженні параметрів моделей адаптивного прогнозування і обчислення прогнозних значень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

– визначення рівня значимості параметра шляхом присвоєння різних рангів;

– перевірку придатності експертних оцінок для подальшого використання;

– визначення оцінки попарного пріоритету параметрів;

– обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 4.3.

Таблиця 4.3 – Результати ранжування параметрів

| Позначення параметра | Назва параметра | Одиниці виміру | Ранг параметра за оцінкою експерта | | | | | | | Сума рангів $R_i$ | Відхилення $\Delta_i$ | $\Delta_i^2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | |
| *X1* | Швидкодія мови програмування | Оп/мс | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 17 | 0,33 | 0,109 |
| *X2* | Час обробки запитів | Мс | 2 | 2 | 1 | 3 | 1 | 1 | 1 | 11 | -14,67 | 215,2 |
| *X3* | Потенційний об’єм програмного коду | кількість строк коду | 1 | 1 | 2 | 1 | 3 | 3 | 3 | 14 | 14,33 | 214,92 |
| | Разом | | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 42 | 0 | 430,229 |

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^{N} r_{ij} R_{ij} = \frac{Nn\ n+1}{2} = 42,$$

де $N$ – число експертів, $n$ – кількість параметрів;

б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 26{,}67$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T$$

Сума відхилень по всім параметрам повинна дорівнювати 0;

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^{N} \Delta_i^2 = 430{,}229$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2\ n^3 - n} = \frac{12 \cdot 430{,}229}{7^2\ 3^3 - 3} = 4{,}39 > W_{\text{норм}} = 0{,}67$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 4.4.

Таблиця 4.4 – Попарне порівняння параметрів

| Параметри | Експерти | | | | | | | Кінцева оцінка | Числове значення |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | |
| X1 і X2 | > | > | > | > | > | > | > | > | 1,5 |
| X1 і X3 | > | > | > | < | > | > | > | > | 0,5 |
| X2 і X3 | > | > | < | > | < | < | < | < | 0,5 |

Числове значення, що визначає ступінь переваги *i*–го параметра над *j*–тим, $a_{ij}$ визначається по формулі:

$$a_{ij} = \begin{matrix} 1.5, \text{при Xi} > X\text{j} \\ 1.0, \text{при Xi} = \text{Xj} \\ 0.5, \text{при Xi} < X\text{j} \end{matrix}$$

З отриманих числових оцінок переваги складемо матрицю A= $\| a_{ij} \|$ .Для кожного параметра зробимо розрахунок вагомості $K_{ві}$ за наступними формулами:

$$K_{\text{ві}} = \frac{b_i}{\sum_{i=1}^{n} b_i},$$

де $b_i = \sum_{i=1}^{N} a_{ij}$.

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%).

На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{\text{ві}} = \frac{b_i'}{\sum_{i=1}^{n} b_i'}, \text{де } b_i' = \sum_{i=1}^{N} a_{ij} b_j$$

Як видно з таблиці 5.5, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно

Таблиця 4.5 – Розрахунок вагомості параметрів

| Параметри $x_i$ | Параметри $x_j$ | | | Перша ітер. | | Друга ітер. | | Третя ітер. | |
|---|---|---|---|---|---|---|---|---|---|
| | X1 | X2 | X3 | $b_i$ | $K_{\text{ві}}$ | $b_i^1$ | $K_{\text{ві}}^1$ | $b_i^2$ | $K_{\text{ві}}^2$ |
| X1 | 1,0 | 1,5 | 0,5 | 3.0 | 0.333 | 8.0 | 0.32 | 21.25 | 0.31 |
| X2 | 0,5 | 1,0 | 0,5 | 2.0 | 0.222 | 5.5 | 0.22 | 15.25 | 0.223 |
| X3 | 1,5 | 1,5 | 1,0 | 4.0 | 0. 445 | 11.5 | 0.46 | 31.75 | 0.465 |
| Всього: | | | | 9 | 1 | 25 | 1 | 68.25 | 1 |

## 4.3 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо. Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 4.6):

$$K_K\ j\ = \sum_{i=1}^{n} K_{\text{в}i,j} B_{i,j},$$

де $n$ – кількість параметрів;

$K_{\text{в}i}$– коефіцієнт вагомості $i$–го параметра;

$B_i$ – оцінка $i$–го параметра в балах.

Таблиця 4.6 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

| Основні функції | Варіант реалізації функції | Параметри $x_i$ | Абсолютне значення параметра | Бальна оцінка параметра | Коефіцієнт вагомості параметра | Коефіцієнт рівня якості |
|---|---|---|---|---|---|---|
| F1 | б) | Xl | 11000 | 7 | 0,312 | 2,185 |
| | | X3 | 200 | 5 | 0,465 | 2,325 |
| F2 | б) | X2 | 100 | 3 | 0,223 | 0,662 |
| | | X3 | 200 | 7,5 | 0,465 | 3,488 |
| F3 | б) | X2 | 100 | 3,5 | 0,78 | 0,78 |
| | | X3 | 350 | 1 | 0,465 | 0,465 |
| | а) | X3 | 200 | 1 | 0,461 | 0,461 |

За даними з таблиці 4.6 за формулою

$$K_K = K_{\text{ТУ}}\ F_{1k}\ + K_{\text{ТУ}}\ F_{2k}\ + \ldots + K_{\text{ТУ}}\ F_{zk}\ ,$$

визначаємо рівень якості кожного з варіантів:

$$К_{К1} = 2,185+2,325+0,662+3,488+0,78+0,465 = 9,891$$

$$К_{К2} = 2,185+2,325+0,662+3,488+0,461 = 9,121$$

Як видно з розрахунків, кращим є перший варіант, для якого коефіцієнт технічного рівня має найбільше значення.

## 4.4 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Всі варіанти включають в себе два окремих завдання:

1. Розробка проекту програмного продукту;
2. Розробка програмної оболонки;

Але варіант II реалізації програмного забезпечення включає ще одне завдання:

3. Написання алгоритму збереження інформації у вигляді компонента, зручного для візуалізації.

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 – до групи Б, завдання 3 до групи Г. За складністю алгоритми, які використовуються в завданні 1 належать до групи 1; а в завданні 2 – до групи 3. Завдання 3 відноситься за складністю до групи 3.

Для реалізації завдання 1 використовується довідкова інформація, а завдання 2 використовує інформацію у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань. Загальна трудомісткість обчислюється як

$$T_O = T_P \cdot K_П \cdot K_{СК} \cdot K_М \cdot K_{СТ} \cdot K_{СТ.М},$$

де $T_P$ – трудомісткість розробки ПП;

$K_П$ – поправочний коефіцієнт;

$K_{СК}$ – коефіцієнт на складність вхідної інформації;

$K_М$ – коефіцієнт рівня мови програмування;

$K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм;

$K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру степеню новизни А та групи складності алгоритму 1, трудомісткість дорівнює: $T_P$ =90 людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання: $K_П$ = 1.7. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для завдань рівний 1: $K_{СК}$ = 1. Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $K_{СТ}$ = 0.8. Тоді, за формулою, загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 90 \cdot 1.7 \cdot 0.8 = 122.4 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для подальших завдань.

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни Б), тобто $T_P$ =27 людино-днів, $K_П$ =0.9, $K_{СК}$ = 1, $K_{СТ}$ =0.8:

$$T_2 = 27 * 0.9 * 0.8 = 19.44 \text{ людино-днів.}$$

Для третього завдання (використовується алгоритм третьої групи складності, ступінь новизни Г):

$T_р$ = 15 людино-днів; $K_п$ = 0,6; $K_{ст}$ = 1; $T_З$ = 15 * 0,6 *1 = 9.

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$T_I = (122.4 + 19.44) * 15 = 2127,6$ людино-годин;

$T_{II} = (122.4 + 19.44 + 9) * 15 = 2262,6$ людино-годин;

Найбільш високу трудомісткість має варіант II.

В розробці беруть участь два програмісти з окладом 14000 грн., один спеціаліст по цифровій обробці сигналів з окладом 22000грн. Визначимо зарплату за годину за формулою:

$$СЧ = \frac{М}{T_m \cdot t} \text{ грн.},$$

де М – місячний оклад працівників;

$T_m$ – кількість робочих днів тиждень;

$t$ – кількість робочих годин в день.

$$СЧ = \frac{14000 + 14000 + 22000}{3 \cdot 21 \cdot 15} = 52,91 \text{ грн.}$$

Тоді, розрахуємо заробітну плату за формулою

$$СЗП = С_ч \cdot Т_i \cdot КД,$$

де $С_ч$ – величина погодинної оплати праці програміста;

$Т_i$ – трудомісткість відповідного завдання;

$К_Д$ – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

I. $С_{ЗП} = 52,91 \cdot 2127,6 \cdot 1.2 = 135085,58$ грн.

II. $С_{ЗП} = 52,91 \cdot 2262,6 \cdot 1.2 = 143657,00$ грн.

Відрахування на єдиний соціальний внесок в залежності від групи професійного ризику (II клас) становить 22%:

I. $С_{ВІД} = С_{ЗП} \cdot 0.3677 = 135085,58 \cdot 0.22 = 29718,83$ грн.

II. $С_{ВІД} = С_{ЗП} \cdot 0.3677 = 143657,00 \cdot 0.22 = 31604,54$ грн.

Тепер визначимо витрати на оплату однієї машино-години. ($С_М$)

Так як одна ЕОМ обслуговує одного програміста з окладом 14000 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$С_Г = 12 \cdot М \cdot К_З = 12 \cdot 12000 \cdot 0,2 = 28800 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$С_{ЗП} = С_Г \cdot (1 + К_З) = 28800 \cdot (1 + 0.2) = 34560 \text{ грн.}$$

Відрахування на єдиний соціальний внесок:

$$С_{ВІД} = С_{ЗП} \cdot 0.3677 = 34560 \cdot 0,22 = 7603,20 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 8000 грн.

$$С_А = К_{ТМ} \cdot К_А \cdot Ц_{ПР} = 1.15 \cdot 0.25 \cdot 8000 = 2300 \text{ грн.},$$

де $К_{ТМ}$– коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;

$К_А$– річна норма амортизації;

$Ц_{ПР}$– договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$С_Р = К_{ТМ} \cdot Ц_{ПР} \cdot К_Р = 1.15 \cdot 8000 \cdot 0.05 = 460 \text{ грн.},$$

де $К_Р$– відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$Т_{ЕФ} = (Д_К - Д_В - Д_С - Д_Р) \cdot t_З \cdot К_В$$

$$Т_{ЕФ} = (365 - 104 - 8 - 16) \cdot 8 \cdot 0.9 = 1706.4 \text{ годин},$$

де $Д_К$ – календарна кількість днів у році;

$Д_В$, $Д_С$ – відповідно кількість вихідних та святкових днів;

$Д_Р$ – кількість днів планових ремонтів устаткування;

t –кількість робочих годин в день;

$К_В$– коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$С_{ЕЛ} = Т_{ЕФ} \cdot N_С \cdot К_З \cdot Ц_{ЕН} = 1706,4 \cdot 0,156 \cdot 0,9733 \cdot 2,7515 = 712,88 \text{ грн.},$$

де $N_С$ – середньо-споживча потужність приладу;

$К_З$– коефіцієнтом зайнятості приладу;

$Ц_{ЕН}$ – тариф за 1 КВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$С_Н = Ц_{ПР} \cdot 0.67 = 8000 \cdot 0,67 = 5360 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$С_{ЕКС} = С_{ЗП} + С_{ВІД} + С_А + С_Р + С_{ЕЛ} + С_Н$$

$$С_{ЕКС} = 34560 + 7603,20 + 2300 + 460 + 712,88 + 5360 = 50996,08 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$С_{М\text{-}Г} = С_{ЕКС} / Т_{ЕФ} = 50996,08/1706,4 = 29,885 \text{ грн/час.}$$

Оскільки в даному випадку всі роботи, які пов‘язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$С_М = С_{М\text{-}Г} \cdot Т$$

I. $С_М = 29,885 \cdot 2127,6 = 63583,326$ грн.;

II. $С_М = 29,885 \cdot 2262,6 = 67617,801$ грн.;

Накладні витрати складають 67% від заробітної плати:

$$С_Н = С_{ЗП} \cdot 0,67$$

I. $С_Н = 135085,58 \cdot 0,67 = 90507,34$ грн.;

II. $С_Н = 143657,00 \cdot 0,67 = 96250,19$ грн.;

Отже, вартість розробки ПП за варіантами становить:

$$С_{ПП} = С_{ЗП} + С_{ВІД} + С_М + С_Н$$

I. $С_{ПП}$ = 135085,58 + 49670,97+ 63583,326 + 90507,34 = 338847,216 грн.;

II. $С_{ПП}$ = 143657,00 + 52822,68 + 67617,801+ 96250,19 = 360347,671 грн.;

## 4.5 Вибір кращого варіанта ПП за техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$К_{ТЕРj} = К_{Кj} / С_{Фj},$$

$К_{ТЕР1}$ = 9,891/ 344325,79 = $2,87 \cdot 10^{-5}$;

$К_{ТЕР2}$ = 9,121/ 366173,97= $2,49 \cdot 10^{-5}$;

Як бачимо, найбільш ефективним є перший варіант реалізації програми з коефіцієнтом техніко-економічного рівня $К_{ТЕР1}$= $2,88 \cdot 10^{-5}$.

## Висновки до розділу 4

В даній розрахунковій роботі проведено повний функціонально-вартісний аналіз ПП, який було розроблено в рамках дипломної роботи. Процес аналізу можна умовно розділити на дві частини.

В першій з них проведено дослідження ПП з технічної точки зору: було визначено основні функції ПП та сформовано множину варіантів їх реалізації; на основі обчислених значень параметрів, а також експертних оцінок їх важливості було обчислено коефіцієнт технічного рівня, який і дав

змогу визначити оптимальну з технічної точки зору альтернативу реалізації функцій ПП.

Другу частину ФВА присвячено вибору із альтернативних варіантів реалізації найбільш економічно обґрунтованого. Порівняння запропонованих варіантів реалізації в рамках даної частини виконувалось за коефіцієнтом ефективності, для обчислення якого були обчислені такі допоміжні параметри, як трудомісткість, витрати на заробітну плату, накладні витрати.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишились після першого відбору двох варіантів виконання програмного комплексу оптимальним є перший варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості $К_{ТЕР}2{,}88 \cdot 10^{-5}$.

Цей варіант реалізації програмного продукту має такі параметри:

- мова програмування – SAS;
- використання SAS Enterprise Guide;
- консольний інтерфейс.

Даний варіант виконання програмного комплексу дає користувачу відмінний функціонал, швидкодію і робить простішим виконання завдання.

# ВИСНОВКИ

Дипломна робота присвячена задачі аналізу, виявленню та прогнозуванню корупційних злочинів, що вчиняються на території України.

За допомогою методів регресійного аналізу був проведений аналіз даних Transparency International та Національного агентства с питань запобігання корупції за щорічними деклараціями державних службовців. В даній роботі проаналізовано щорічні декларації державних службовців про доходи за 2017 рік та основні індекси устрою життя населення.

В першому розділі було надано визначення корупції, надано характеристику сучасної ситуації з корупцією в Україні, також розглянуто програмний продукт SAS та рішення що він пропонує для боротьби зі злочинністю, та прогнозуванню.

У другому розділі приведені математичні методи побудови прогнозних моделей та сучасні методи краулингу.

Третій розділ присвячено опису розробленої в рамках дипломної роботи системи, показано її застосування на реальних даних.

Четвертий розділ присвячено функціонально-вартісному аналізу.

Новизною даної роботи є те, що було створено інформаційно-аналітичну систему котра б обробляла декларації, написано модуль для аналізу кореляції рядів із використанням мови програмування python.

Запропоновану методику випробувано на реальних деклараціях.

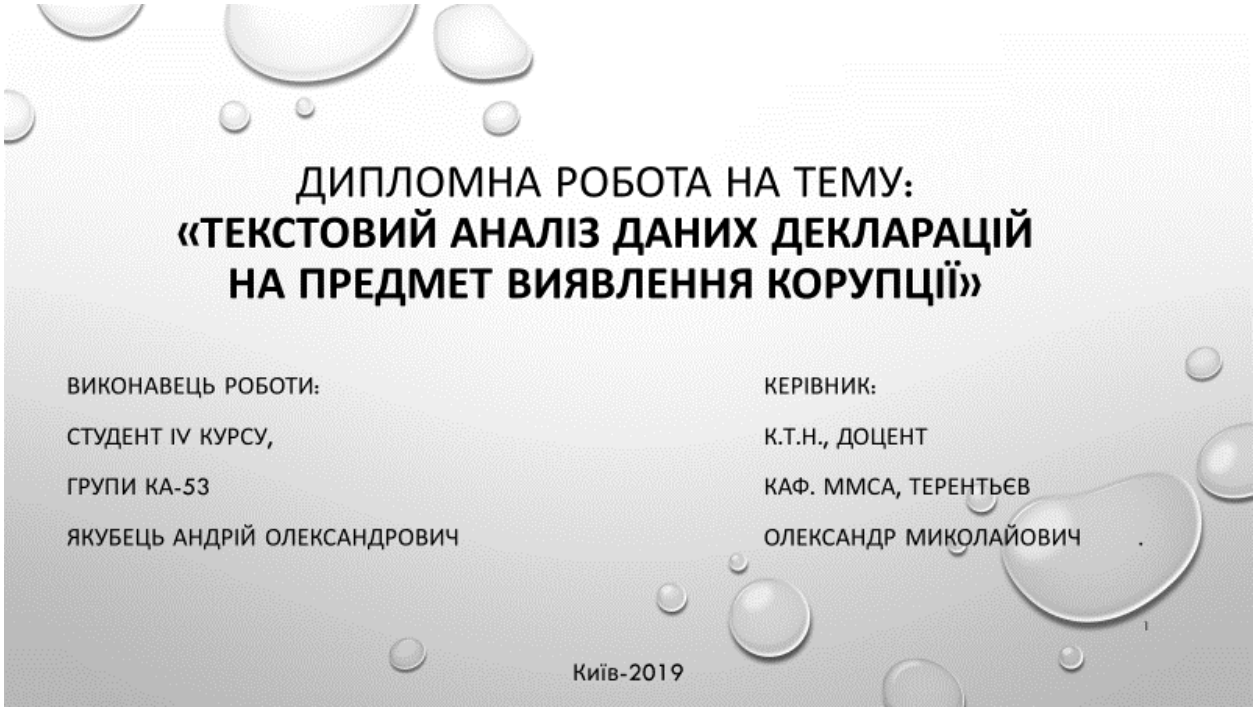
На основі дипломної роботи були написані тезиси та наукова стаття, що приймали участь у наукових конференціях.

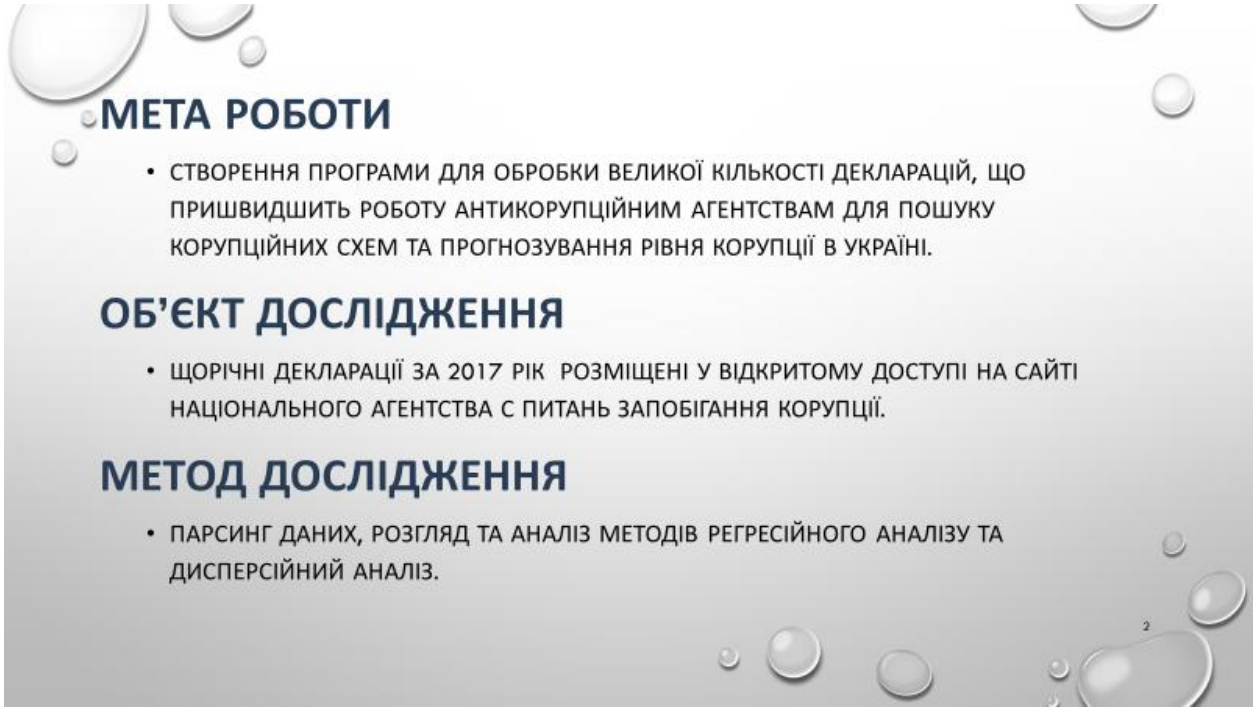
# ПЕРЕЛІК ПОСИЛАНЬ


1. Кохан Г.В. Явище політичної корупції: теоретико-методологічний аналіз. Монографія. Київ.: НІСД, 2013. 232 с.
2. Матеріали науково-практичної конференції (19 квітня 2013 року, м. Харків). МВС України, Харківський національний університет внутрішніх справ, Кримінологічна асоціація України. Харків: Золота миля, 2013. 322 с.
3. Курило Т.С. Адміністративно - правові засади протидії корупції в сфері здійснення нотаріальної діяльності в Україні. Дисертація кандидата юридичних наук: 12.00.07; Науково дослідний інститут публічного права. Київ, 2017. 258 с.
4. Згідно з дослідженням EY, керівникам не вдається ефективно формувати принципи ділової етики. Електронний ресурс, http://www.ey.com/ua/uk/newsroom/news-releases/news-ey-senior-managers-failing-to-set-right-tone-on-business-ethics-finds-ey-fraud-survey
5. Бокс Дж., Дженкинс Г. Анализ временных рядов. Прогноз и управление. М.: Мир, 1974. Вып. 1, 2.
6. Терентьев А.Н., Домрачев В.М., Костецкий Р.И. SAS Base: Основы программирования. Эдельвейс, 2014. 304 с
7. Бідюк П.І., Романенко В.Д., Тимощук О.Л. Аналіз часових рядів: навчальний посібник. К: Політехніка, 2010. 317 с.
8. Ярушкина Н.Г., Афанасьева Т.В. Интеллектуальный анализ временных рядов. Учебное пособие. Ульяновск: УлГТУ, 2010. 320 с. ISBN 978-5-9795-0618-0.
9. Бриллинджер Д. Временные ряды. Обработка данных и теория. М.: Мир, 1980. 536 с.
10. Носко В.П. Эконометрика: Введение в регрессионный анализ временных рядов. Г.Москва, 2002г., 254 стр.

11. Vishal M., Sabri S. Machine Learning for Humans. URL: https:/medium.com/machine-learning-for-human (дата звернення 12.02.2018).
12. Goodfellow I., Bengio Y., Courville A. Deep Learning (Adaptive Computation and Machine Learning series). Cambridge, MA: MIT Press, 2017. 775 p.
13. Friedl Jeffrey E.F. Mastering Regular Expressions. 3rd edition. O'Reilly Media, 2006. 535 p. ISBN: 978-0-596-52812-4
14. Митчелл Райан. Скрапинг веб-сайтов с помощью Python. Пер. с англ. М.: ДМК Пресс, 2016. 280 с. ISBN 978149191029.
15. Zhu X. Semi-Supervised Learning. Elsevier: Academic Press Library in Signal Processing, 2014. 10 p.
16. Graupe D. Principles of artificial neural networks. World Scientific Publishing Co. Pte. Ltd.: Singapore, 2007. 329 p.
17. Мак-Каллок У.С., Питтс В. Логическое исчисление идей, относящихся к нервной активности. ред. К.Э. Шеннона и Дж. Маккарти. Москва: Издательство иностранной литературы, 1956. с.363–384.
18. Rosenblatt R. Principles of Neurodynamics. New York: Spartan Books, 1962. 457 p.
19. Минский М., Пейперт С. Персептроны. Москва: Мир, 1971. 261 с.
20. Кононюк А. Нейронні мережі і генетичні алгоритми. Київ: Корнійчук, 2008. 446 с.
21. Krose B., van der Smagt P. An introduction to Neural Networks. Amsterdam: The University of Amsterdam, 1996. 135 p.

# ДОДАТОК А ІЛЮСТРАТИВНІ МАТЕРІАЛИ ДО ДОПОВІДІ



## АКТУАЛЬНІСТЬ РОБОТИ

- В УКРАЇНІ ПОСТІЙНО ВІДБУВАЮТЬСЯ РЕФОРМИ У ВСІХ СФЕРАХ ДЕРЖАВНОГО АПАРАТУ. ОДНОЮ ІЗ ТАКИХ РЕФОРМ – Є АНТИКОРУПЦІЙНА РЕФОРМА. СТВОРЕННЯ СИСТЕМИ, ЯКА ЗМОЖЕ ВИЯВЛЯТИ ДЕРЖАВНИХ СЛУЖБОВЦІВ ЗА ПРАВИЛАМИ РИЗИКУ ДОПОМОЖЕ ВИКОРІНИТИ КОРУПЦІЮ З НАЙВИЩИХ ЛАНОК.
- ЗА ЗАКОНОМ ПРО ЗАПОБІГАННЯ КОРУПЦІЇ — ВСІ ДЕРЖАВНІ СЛУЖБОВЦІ ЗОБОВ'ЯЗАНІ ЗАПОВНИТИ ТА ОПРИЛЮДНИТИ ДЕКЛАРАЦІЇ МАЙНОВОГО СТАНУ ДЛЯ ВІЛЬНОГО ДОСТУПУ ТА ЇХ ПЕРЕВІРКИ. ПОЧИНАЮЧИ З 2015 РОКУ, НА САЙТІ НАЦІОНАЛЬНОГО АГЕНТСТВА С ПИТАНЬ ЗАПОБІГАННЯ КОРУПЦІЇ З'ЯВЛЯЄТЬСЯ БІЛЬШЕ 1 МЛН. ДЕКЛАРАЦІЙ.

3

## ПОСТАНОВКА ЗАДАЧІ

- РОЗРОБИТИ ПРОГРАМУ ДЛЯ ПАРСИНГУ, ОБРОБКИ ТА ЗАВАНТАЖЕННЯ ДАНИХ У ІНФОРМАЦІЙНО-АНАЛІТИЧНУ СИСТЕМУ;
- РОЗРОБИТИ СТРУКТУРИ ТАБЛИЦІ ЗНАЧЕНЬ РИЗИКІВ ДЕКЛАРАНТА, ЗА ЗАДАНИМИ ПРАВИЛАМИ РИЗИКУ ТА КРИТЕРІЯМИ ВІДБОРУ;
- РОЗРОБИТИ ПРОГРАМУ ДЛЯ ПРОГНОЗУВАННЯ КОРУПЦІЇ В УКРАЇНІ ТА ПРОВЕСТИ АНАЛІЗ НА СТАТИСТИЧНУ ЗНАЧИМІСТЬ РЕЗУЛЬТАТІВ ПРОГНОЗУ;
- ПРОТЕСТУВАТИ ПРОГРАМУ НА ЗАВАНТАЖЕНИХ ІЗ САЙТУ РЕАЛЬНИХ ДАНИХ.

4

# СТАН І ТЕНДЕНЦІЇ КОРУПЦІЇ В УКРАЇНІ

- У РЕЙТИНГУ ТІ УКРАЇНА СТАЛА НАЙКОРУМПОВАНІШОЮ КРАЇНОЮ ЄВРОПИ. У ДОСЛІДЖЕННІ ERNST&YOUNG ЗА 2017 РІК УКРАЇНА ЗАЙНЯЛА ПЕРШЕ МІСЦЕ ЗА ПОШИРЕНІСТЮ ХАБАРНИЦТВА/КОРУПЦІЇ У ДІЛОВІЙ ПРАКТИЦІ

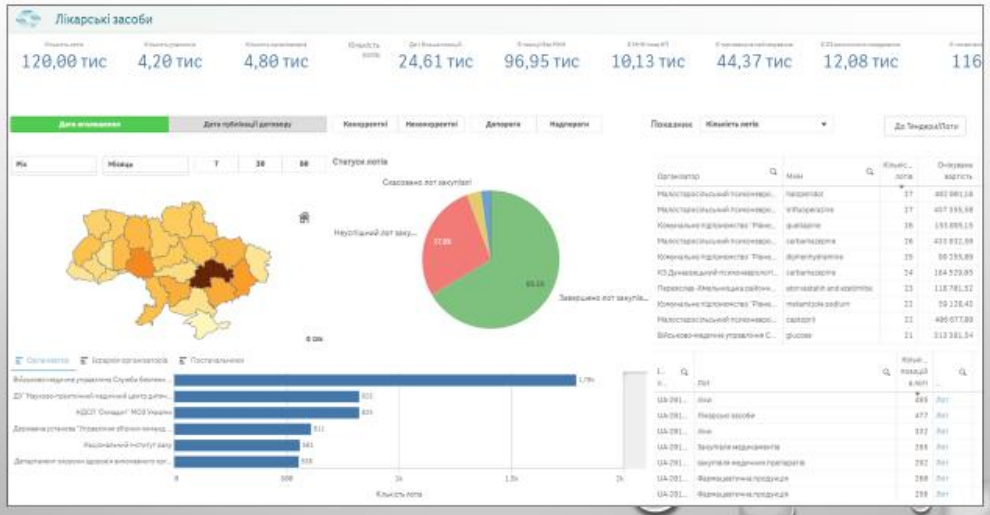
TRANSPARENCY INTERNATIONAL
the global coalition against corruption


5

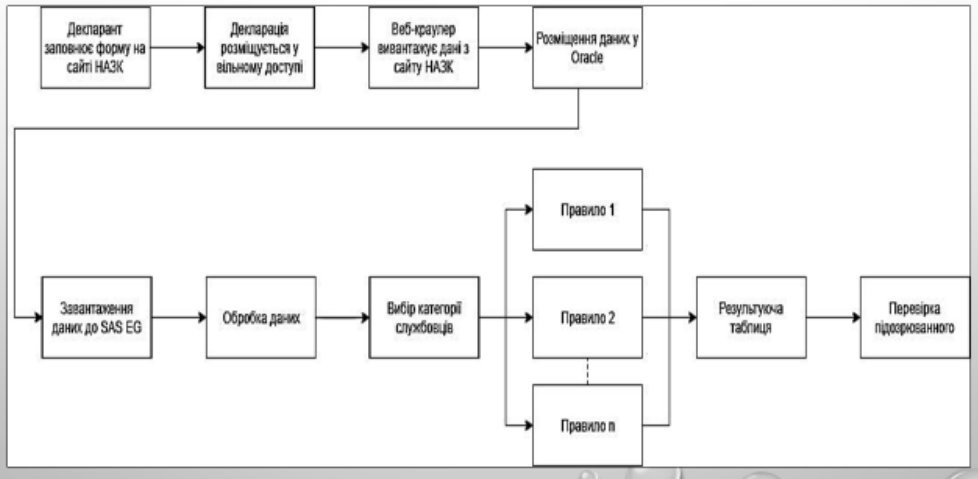
# ОГЛЯД SAS® MEMEX


6

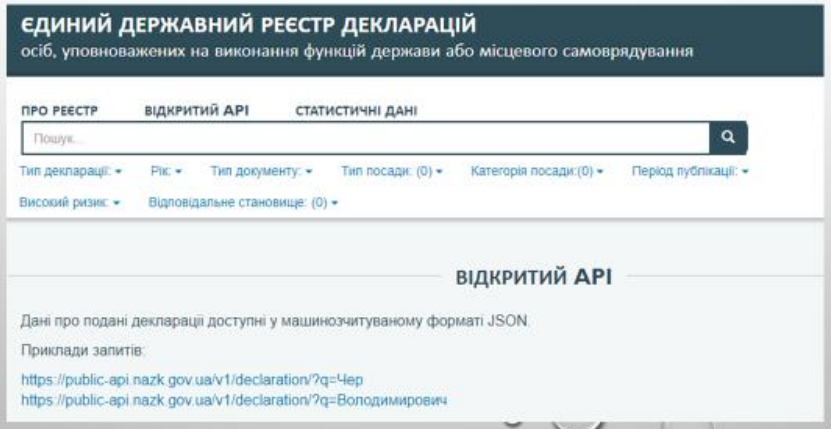
# ОГЛЯД QLIK® SENSE




# АРХІТЕКТУРА ПОБУДОВИ КАРТИ РИЗИКУ



# ПАРСИНГ ДАНИХ З САЙТУ НАЗК



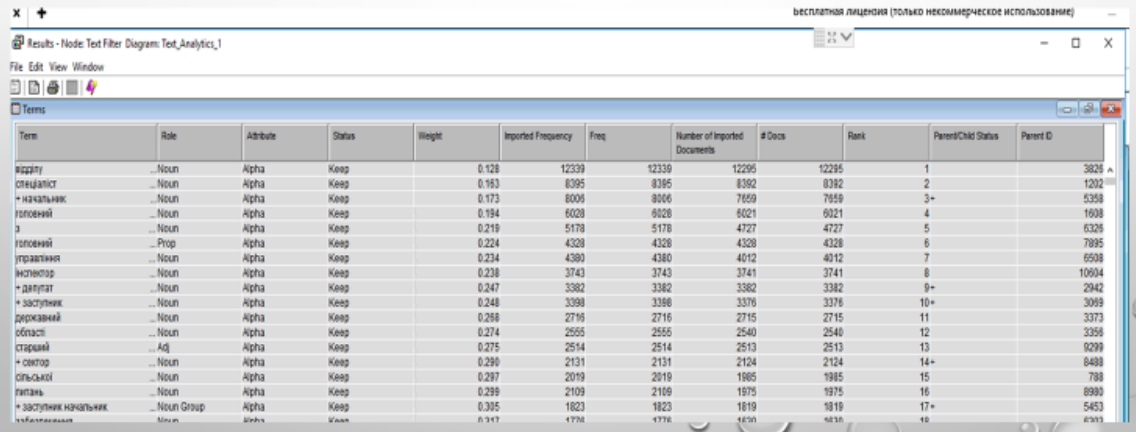



# ВІДПОВІДЬ НА ЗАПИТ ПАРСЕРА

- [ { "ID":"B238B773-316F-49E4-84A5-AEA634E18B65",
  "CREATED_DATE":"05.09.2016",
  "LASTMODIFIED_DATE":"05.09.2016",
  "DATA":{
  "STEP_0":{
  "DECLARATIONTYPE":"1",
  "DECLARATIONYEAR1":"2015"

…}]

# КЛАСТЕРИЗАЦІЯ ДЕКЛАРАНТІВ ЗА ПОСАДОЮ

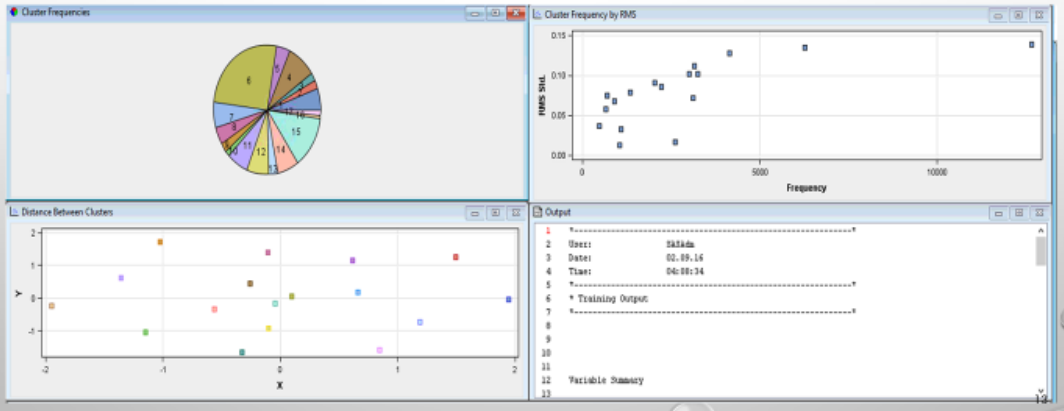



| Term | Role | Attribute | Status | Weight | Imported Frequency | Freq | Number of Imported Documents | # Docs | Rank | Parent/Child Status | Parent ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| відділу | Noun | Alpha | Keep | 0.128 | 12339 | 12339 | 12295 | 12295 | 1 | | 3826 |
| спеціаліст | Noun | Alpha | Keep | 0.163 | 8395 | 8395 | 8392 | 8392 | 2 | | 1202 |
| + начальник | Noun | Alpha | Keep | 0.173 | 8006 | 8006 | 7659 | 7659 | 3+ | | 5358 |
| головний | Noun | Alpha | Keep | 0.194 | 6028 | 6028 | 6021 | 6021 | 4 | | 1608 |
| з | Noun | Alpha | Keep | 0.219 | 5178 | 5178 | 4727 | 4727 | 5 | | 6326 |
| головний | Prop | Alpha | Keep | 0.224 | 4328 | 4328 | 4328 | 4328 | 6 | | 7895 |
| управління | Noun | Alpha | Keep | 0.234 | 4380 | 4380 | 4012 | 4012 | 7 | | 6508 |
| інспектор | Noun | Alpha | Keep | 0.238 | 3743 | 3743 | 3741 | 3741 | 8 | | 10604 |
| + депутат | Noun | Alpha | Keep | 0.247 | 3382 | 3382 | 3382 | 3382 | 9+ | | 2942 |
| + заступник | Noun | Alpha | Keep | 0.248 | 3398 | 3398 | 3376 | 3376 | 10+ | | 3069 |
| державний | Noun | Alpha | Keep | 0.268 | 2716 | 2716 | 2715 | 2715 | 11 | | 3373 |
| області | Noun | Alpha | Keep | 0.274 | 2555 | 2555 | 2540 | 2540 | 12 | | 3356 |
| старший | Adj | Alpha | Keep | 0.275 | 2514 | 2514 | 2513 | 2513 | 13 | | 9299 |
| + сектор | Noun | Alpha | Keep | 0.290 | 2131 | 2131 | 2124 | 2124 | 14+ | | 8488 |
| сільської | Noun | Alpha | Keep | 0.297 | 2019 | 2019 | 1985 | 1985 | 15 | | 788 |
| питань | Noun | Alpha | Keep | 0.299 | 2109 | 2109 | 1975 | 1975 | 16 | | 8980 |
| + заступник начальник | Noun Group | Alpha | Keep | 0.305 | 1823 | 1823 | 1819 | 1819 | 17+ | | 5453 |

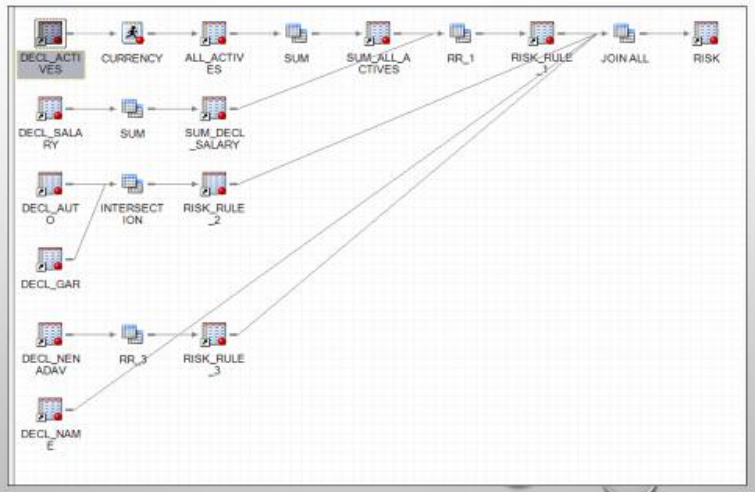
# РЕЗУЛЬТАТИ КЛАСТЕРИЗАЦІЇ

| Cluster ID | Descriptive Terms | Frequency |
|---|---|---|
| 4.0 | відділення +командир служби військовослужбовець україни водій частини пожежний-рятувальник | 4155.0 |
| 12.0 | головний спеціаліст державний з інспектор питань призначення пенсій перерахунку осіб зборів п | 3260.0 |
| 14.0 | головний +бухгалтер спеціаліст державний ревізор-інспектор інспектор лікар +головной +'головно | 3160.0 |
| 11.0 | +начальник відділу +заступник +'заступник начальник' дільниці станції +'начальник караул' +караул | 3125.0 |
| 7.0 | секретар голова суддя +судовой засідання сільський сільської помічник +суд судді засідань судови | 3018.0 |
| 1.0 | депутат сільської селищної скликання працюю вчитель сільський ради підприємець приватний учи | 2625.0 |
| 8.0 | старший оперуповноважений слідчий інспектор державний офіцер майстер виконавець особливо | 2220.0 |
| 5.0 | поліції поліцейський інспектор патрульної +сектор реагування офіцер дільничний роти батальйону | 2048.0 |
| 13.0 | інспектор молодший відділу охорони і безпеки нагляду +режим категорії публічної інспектор-кінс | 1348.0 |

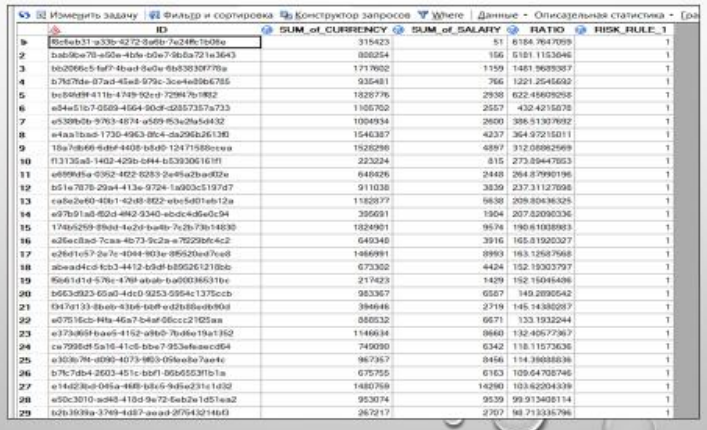
# РЕЗУЛЬТАТИ КЛАСТЕРИЗАЦІЇ




# СХЕМА ПРОЦЕСУ КАРТИ РИЗИКУ



# КАРТА РИЗИКУ ДЕКЛАРАЦІЙ



Измерить задачу | Фильтр и сортировка | Конструктор запросов | Where | Данные | Описательная статистика | Гра

| | ID | SUM_of_CURRENCY | SUM_of_SALARY | RATIO | RISK_RULE_1 |
|---|---|---|---|---|---|
| 1 | f8c6eb31-a33b-4272-8e6b-7e24ffc1b08e | 315423 | 51 | 6184.7647059 | 1 |
| 2 | bab9be78-e60e-4bfe-b0e7-9b8a721e3643 | 886254 | 156 | 5181.1153846 | 1 |
| 3 | bb2066c5-faf7-4bad-8e0e-6b83830f778a | 1717602 | 1159 | 1481.9689387 | 1 |
| 4 | b7fd7fde-87ad-45e8-979c-3ce4e89b6785 | 935481 | 766 | 1221.2545692 | 1 |
| 5 | bc84fd9f-411b-4749-92cd-72947b1f82 | 1828776 | 2938 | 622.45609258 | 1 |
| 6 | e84e51b7-0589-4564-90df-d2857357a733 | 1105702 | 2557 | 432.4215878 | 1 |
| 7 | e539b0b-9763-4874-a589-f53e2fa5d432 | 1004934 | 2600 | 386.51307692 | 1 |
| 8 | e4aa1bad-1730-4963-8fc4-da296b26138 | 1546387 | 4237 | 364.97215011 | 1 |
| 9 | 18a7db66-6dbf-4408-b8d0-12471588ccea | 1528298 | 4897 | 312.08862569 | 1 |
| 10 | f13135a8-1402-429b-bf44-b539306161f1 | 223224 | 815 | 273.89447853 | 1 |
| 11 | e699d5a-0362-4f22-8283-2e49a2bad02e | 648426 | 2448 | 264.87990196 | 1 |
| 12 | b51e7878-29a4-413e-9724-1a903c5197d7 | 911038 | 3839 | 237.31127898 | 1 |
| 13 | ca8e2e60-46b1-42d8-8f22-ebe5d01eb12a | 1182877 | 5638 | 209.80436325 | 1 |
| 14 | e97b91a8-f82d-4f42-9340-ebdc4d6e0c94 | 395691 | 1904 | 207.82093336 | 1 |
| 15 | 174bb259-89dd-4e2d-ba4b-7c2b73b14830 | 1824901 | 9574 | 190.61008883 | 1 |
| 16 | e26ec8ad-7caa-4b73-9c2a-e75229bfc4c2 | 640348 | 3916 | 165.81920327 | 1 |
| 17 | e26d1c67-2e7e-4044-903e-8f562bed7ce8 | 1466991 | 8993 | 163.12687568 | 1 |
| 18 | abead4cd-fcb3-4412-b9df-b89526121bbb | 673302 | 4424 | 152.18003797 | 1 |
| 19 | f6b61d1d-578c-47bf-abab-ba00036531bc | 217423 | 1429 | 152.15046486 | 1 |
| 20 | b663d923-65a0-4dc0-9253-5954c1375ccb | 983367 | 6587 | 149.2890542 | 1 |
| 21 | 0d7d133-8beb-43b6-bbff-ed2b88edb80d | 394646 | 2719 | 145.14380287 | 1 |
| 22 | e07516cb-f4fa-46a7-b4af-06ccc21f25aa | 886532 | 6671 | 133.1932244 | 1 |
| 23 | e373d69f-bae5-4152-a9b0-7bd6e19a1362 | 1146634 | 8660 | 132.40577367 | 1 |
| 24 | ce7998df-5a16-41c6-bbe7-953efeaecd64 | 749090 | 6342 | 118.11573636 | 1 |
| 25 | e303b7ff-d090-4073-9f03-05fee8e7ae4c | 967357 | 8466 | 114.28888836 | 1 |
| 26 | b7fc7db4-2603-451c-bbf1-86b6553f1b1a | 675755 | 6163 | 109.64708746 | 1 |
| 27 | e14d23bd-045a-46f8-b8c6-9d5e231c1d32 | 1480759 | 14290 | 103.62204339 | 1 |
| 28 | e50c3010-ad48-418d-9e72-6eb2e1d51ea2 | 953074 | 9539 | 99.913408114 | 1 |
| 29 | b2b3939a-3769-4d87-aead-2f7643214b63 | 267217 | 2707 | 98.713335796 | 1 |

15

# ПОБУДОВА ПРОГНОЗУЮЧОЇ МОДЕЛІ

| | | | |
|---|---|---|---|
| **Індекс Сприйняття корупції** | CPI | Вимірює, наскільки поширена корупція в державному секторі даної країни. Країни оцінені від 0 до 10, де 10 – найменш корумпована країна. | Рейтинг |
| **ВВП** | GDPpc | Вартість товарів і послуг, вироблених в одній країні поділений на населення даної країни. | Долари США |
| **Рівень безробіття** | UnempRate | Процент безробітних серед робочої сили. | Процент |
| **Рівень інфляції** | InflaRate | Темпи зростання споживчих цін. | Процент |

16

# РЕЗУЛЬТАТИ ПОБУДОВИ

$$CPI = B_0 + B_1\,GDP - B_2\,UNEMPRATE - B_3 INFLARATE + U_i$$

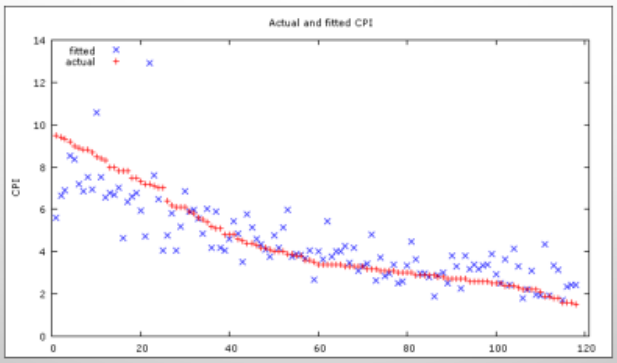
| Source | SS | df | MS | Number of obs = 118 |
|---|---|---|---|---|
| Model | 405.197884 | 3 | 135.065961 | F( 3, 114) = 84.75 |
| Residual | 181.676347 | 114 | 1.59365217 | Prob > F = 0.0000 |
| | | | | R-squared = 0.6904 |
| | | | | Adj R-squared = 0.6823 |
| Total | 586.874231 | 117 | 5.01601907 | Root MSE = 1.2624 |

| CPI | Coef. | Std. Err. | t | P>\|t\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| GDPpc | .0000918 | 8.02e-06 | 11.46 | 0.000 | .0000759 | .0001077 |
| UnempRate | -.0112513 | .0082036 | -1.37 | 0.173 | -.0275025 | .0049999 |
| InflaRate | -.1030416 | .030841 | -3.34 | 0.001 | -.1641375 | -.0419458 |
| _cons | 3.533943 | .3461715 | 10.21 | 0.000 | 2.84818 | 4.219706 |

Fig A[1]

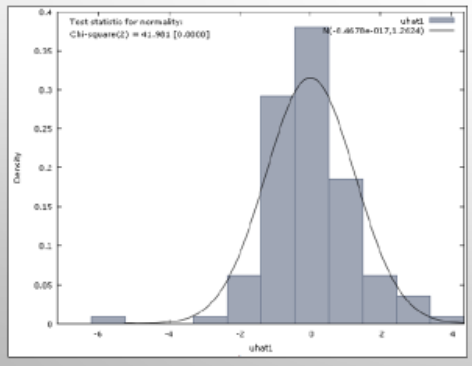
17

# ГРАФІК ЗАЛЕЖНОСТІ МІЖ РЕАЛЬНИМИ ЗНАЧЕННЯМ ТА ЗМОДЕЛЬОВАНИМИ




18

# ОЦІНКА РЕЗУЛЬТАТІВ



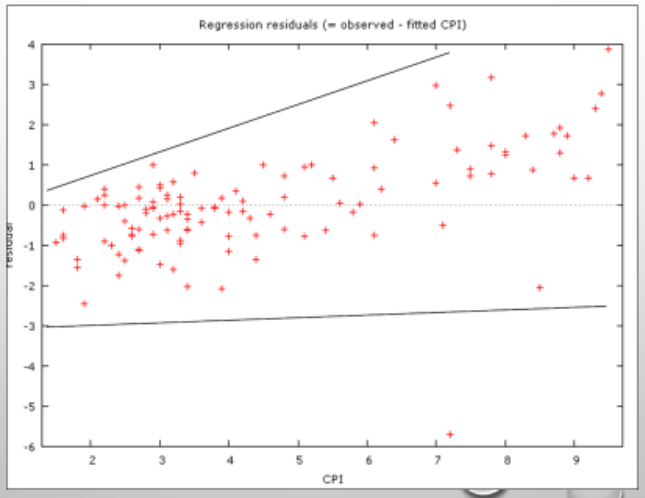

| | Sum of squares | df | Mean square |
|---|---|---|---|
| **Regression** | 405.198 | 3 | 135.066 |
| **Residual** | 181.676 | 114 | 1.59365 |
| **Total** | 586.874 | 117 | 5.01602 |

R^2 = 405.198 / 586.874 = 0.690434

F(3, 114) = 135.066 / 1.59365 = 84.7525 [p-value 6.72e-029]

19

# ГРАФІК ЗАЛИШКІВ ВІДНОСНО CPI




20

# ВИСНОВКИ

- РОЗРОБЛЕНО ПРОГРАМУ ДЛЯ ПАРСИНГУ, ОБРОБКИ ТА ЗАВАНТАЖЕННЯ ДАНИХ У ІНФОРМАЦІЙНО-АНАЛІТИЧНУ СИСТЕМУ;
- РОЗРОБЛЕНО СТРУКТУРУ ТАБЛИЦІ ЗНАЧЕНЬ РИЗИКІВ ДЕКЛАРАНТА, ЗА ЗАДАНИМИ ПРАВИЛАМИ РИЗИКУ ТА КРИТЕРІЯМИ ВІДБОРУ;
- РОЗРОБЛЕНО ПРОГРАМУ ДЛЯ ПРОГНОЗУВАННЯ КОРУПЦІЇ В УКРАЇНІ ТА ПРОВЕСТИ АНАЛІЗ НА СТАТИСТИЧНУ ЗНАЧИМІСТЬ РЕЗУЛЬТАТІВ ПРОГНОЗУ;
- ПРОТЕСТОВАНО ПРОГРАМУ НА ЗАВАНТАЖЕНИХ ІЗ САЙТУ РЕАЛЬНИХ ДАНИХ.

21

# ПЕРСПЕКТИВИ ЩОДО ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

- СТВОРЕННЯ ЗВ'ЯЗКУ МІЖ ДЕКЛАРАЦІЯМИ ОДНОГО І ТОГО САМОГО ДЕКЛАРАНТА ЗА РІЗНІ РОКИ
- СТВОРЕННЯ ПЗ ДЛЯ АВТОМАТИЧНОГО ОНОВЛЕННЯ БАЗИ ДАНИХ ДЕКЛАРАЦІЙ
- ДОДАННЯ НОВИХ ПРАВИЛ ДЛЯ КАРТИ РИЗИКУ
- ВПРОВАДЖЕННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ПОШУКУ СЛІДІВ КОРУПЦІЙНОГО ЗЛОЧИНУ

22

# ПУБЛІКАЦІЇ ТА УЧАСТЬ У КОНФЕРЕНЦІЯХ

- ТЕРНЕТЬЄВ О.М., ЯКУБЕЦЬ А.О., ПРОСЯНКІНА-ЖАРОВА Т.І. **ЗАСТОСУВАННЯ SAS ENTERPRISE GUIDE ДЛЯ ВИЯВЛЕННЯ ЗВ'ЯЗКІВ У ПРЕДМЕТНО-ОРІЄНТОВАНИХ ДАНИХ** ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: ТЕОРІЯ І ПРАКТИКА: ТЕЗИ ДОПОВІДЕЙ II ВСЕУКРАЇНСЬКОЇ ІНТЕРНЕТ-КОНФЕРЕНЦІЇ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ І МОЛОДИХ УЧЕНИХ (4 КВІТНЯ 2019 Р., М. ЗАПОРІЖЖЯ)

23

# ДЯКУЮ ЗА УВАГУ!

24

# ДОДАТОК Б ТЕКСТ ПРОГРАМИ

## 1. Програмний код парсингу даних з сайту НАЗК

```
import requests
import json
from time import sleep
from os import listdir
from os.path import isfile, join

mypath = 'C:\\workshop_NABU\\decl\\'
onlyfiles = [f for f in listdir(mypath) if isfile(join(mypath, f))]



decl_id = []
with open('C:\\workshop_NABU\\declaration_links.txt') as f:
   decl_id = f.read().splitlines()
decl_id = decl_id[len(onlyfiles)+1000000:1500000]

for page in range(500000):

        print("Fetching decl #%s" % str(page+len(onlyfiles)+1))
        try:
                subr = requests.get(
                        "https://declarations.com.ua/declaration/nacp_" + decl_id[page]
+
                        "?q=&format=opendata").json()
                with open(mypath + decl_id[page] + '.json', 'w') as outfile:
                        json.dump(subr, outfile)
        except:
```

```
            f = open("bad.txt", "a")
            f.write(decl_id[page] + '\n')
            f.close()
```

**2. Програмний код реалізації правил, щодо виявлення можливих корупційних злочинів**

```
DATA WORK.decl_auto;
  LENGTH
    ID              $ 36
    AUTO_COUNT      8 ;
  FORMAT
    ID              $CHAR36.
    AUTO_COUNT      BEST1. ;
  INFORMAT
    ID              $CHAR36.
    AUTO_COUNT      BEST1. ;
  INFILE 'C:\workshop\NABU\decl_auto.txt'
    LRECL=38
    ENCODING="UTF-8"
    TERMSTR=CRLF
    DLM='7F'x
    MISSOVER
    DSD ;
  INPUT
    ID              : $CHAR36.
    AUTO_COUNT      : BEST1. ;
RUN;
%let USD=28.29;
%let UAH=1.0;
%let EUR=33.13;
%let CHF=29.13;
```

```
%let CNY=4.11;
%let JPY=0.25;
%let SEC=3.21;
%let RUB=0.43;
%let CAN=21.65;
data WORK.ALL_ACTIVES (DROP=CURRENCY1);
set FD.decl_actives;
if CURRENCY1='USD' then CURRENCY=actives*&USD;
else if CURRENCY1='EUR' then CURRENCY=actives*&EUR;
else if CURRENCY1='CNY' then CURRENCY=actives*&CNY;
else if CURRENCY1='RUB' then CURRENCY=actives*&RUB;
else if CURRENCY1='CHF' then CURRENCY=actives*&CHF;
else if CURRENCY1='SEC' then CURRENCY=actives*&SEC;
else if CURRENCY1='JPY' then CURRENCY=actives*&JPY;
else if CURRENCY1='CAD' then CURRENCY=actives*&CAN;
else CURRENCY=actives*&UAH; /*UAH*/
run;
%_eg_conditional_dropds(WORK.SUM_DECL_SALARY);
PROC SQL;
  CREATE TABLE WORK.SUM_DECL_SALARY AS
  SELECT t1.ID,
        /* SUM_of_SALARY */
         (SUM(t1.SALARY)) AS SUM_of_SALARY
    FROM FD.DECL_SALARY t1
    WHERE t1.ID NOT = ''
    GROUP BY t1.ID;
QUIT;

%_eg_conditional_dropds(WORK.SUM_ALL_ACTIVES);
```

```
PROC SQL;
  CREATE TABLE WORK.SUM_ALL_ACTIVES AS
  SELECT t1.ID,
       /* SUM_of_CURRENCY */
        (SUM(t1.CURRENCY)) AS SUM_of_CURRENCY
    FROM WORK.ALL_ACTIVES t1
    WHERE t1.ID NOT = ''
    GROUP BY t1.ID;
QUIT;
%_eg_conditional_dropds(WORK.RISK_RULE_1);

PROC SQL;
  CREATE TABLE WORK.RISK_RULE_1 AS
  SELECT t1.ID,
       t1.SUM_of_CURRENCY,
       t2.SUM_of_SALARY,
       /* RATIO */
        (t1.SUM_of_CURRENCY/t2.SUM_of_SALARY) AS RATIO,
       /* RISK_RULE_1 */
        (CASE
          WHEN (t1.SUM_of_CURRENCY/t2.SUM_of_SALARY > 20)
          THEN 1
          ELSE 0
        END) AS RISK_RULE_1
    FROM WORK.SUM_ALL_ACTIVES t1
       LEFT JOIN WORK.SUM_DECL_SALARY t2 ON (t1.ID = t2.ID)
    ORDER BY RISK_RULE_1 DESC,
          RATIO DESC;
QUIT;
%_eg_conditional_dropds(WORK.RISK_RULE_1);
```

```
PROC SQL;
  CREATE TABLE WORK.RISK_RULE_1 AS
  SELECT t1.ID,
        t1.SUM_of_CURRENCY,
        t2.SUM_of_SALARY,
        /* RATIO */
         (t1.SUM_of_CURRENCY/t2.SUM_of_SALARY) AS RATIO,
        /* RISK_RULE_1 */
         (CASE
           WHEN (t1.SUM_of_CURRENCY/t2.SUM_of_SALARY > 20)
           THEN 1
           ELSE 0
          END) AS RISK_RULE_1
    FROM WORK.SUM_ALL_ACTIVES t1
         LEFT JOIN WORK.SUM_DECL_SALARY t2 ON (t1.ID = t2.ID)
    ORDER BY RISK_RULE_1 DESC,
             RATIO DESC;
QUIT;
%_eg_conditional_dropds(WORK.RISK_RULE_3);

PROC SQL;
  CREATE TABLE WORK.RISK_RULE_3 AS
  SELECT t1.ID,
        t1.NOT_PROVIDE,
        /* RISK_RULE_3 */
         (CASE
           WHEN t1.NOT_PROVIDE^=0
           THEN 1
           ELSE 0
```

```
        END) AS RISK_RULE_3
    FROM FD.DECL_NENADAV t1
    ORDER BY RISK_RULE_3 DESC,
          t1.NOT_PROVIDE DESC;
QUIT;
%_eg_conditional_dropds(FD.RISK);

PROC SQL;
  CREATE TABLE FD.RISK AS
  SELECT t1.ID,
       t4.SURNAME,
       t4.NAME,
       t1.SUM_of_CURRENCY,
       t1.SUM_of_SALARY,
       t1.RATIO,
       t2.AUTO,
       t2.GARAGE_COUNT,
       t3.NOT_PROVIDE,
       t1.RISK_RULE_1,
       t2.RISK_RULE_2,
       t3.RISK_RULE_3,
       /* SUM_OF_RISKS */
        (t1.RISK_RULE_1+t2.RISK_RULE_2+t3.RISK_RULE_3)          AS
SUM_OF_RISKS
    FROM WORK.RISK_RULE_1 t1
       LEFT JOIN WORK.RISK_RULE_2 t2 ON (t1.ID = t2.ID)
       LEFT JOIN WORK.RISK_RULE_3 t3 ON (t1.ID = t3.ID)
       LEFT JOIN FD.DECL_NAME t4 ON (t1.ID = t4.ID)
    ORDER BY SUM_OF_RISKS DESC,
          t1.RISK_RULE_1 DESC,
```

```
        t2.RISK_RULE_2 DESC,
```

**3. Програмний код реалізації моделі прогнозування корупції**

```
import pandas as pd
import numpy as np
import re
from math import exp
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
"""
define the functions may be used
"""
# Define local weighted linear regression function
# lower-case x is the point where the weight is the biggest
# upper-case X is the training data Y is the label of the training data
# if LS is TRUE we calculate the least squares regression (without wheughts)
def lwl_regression(x, X, Y, LS=False, k=1.0):
    ones = np.ones(len(X))
    X = np.array([X.T[0],X.T[1],ones]).T
    m = len(X)
    W = np.matrix(np.zeros((m,m)))
    x = np.array([x[0][0],x[0][1],1])
    for i in range(m):
        xi = np.array(X[i])
        if not LS:
            W[i, i] = exp(np.linalg.norm(x - xi) / (-2 * k ** 2))
        else:
            W[i, i] = 1
    xwx = X.T * W * X
```

```
    if np.linalg.det(xwx) == 0:
        print('xWx is a singular matrix')
        return
    w = xwx.I * X.T * W * Y
    return w

# how to use the trained model
def h(X,w):
    ones = np.ones(len(X))
    X = np.array([X.T[0], X.T[1], ones]).T
    return X * w

# build training set and testig set
def dataset(X):
    train = X.sample(int(len(X) / 1.5))
    test = X.drop(train.index.values)
    return train, test

bond = pd.read_csv("./data/data.csv")
bond = bond.values
final_data = []
for j in range(len(bond) - 2):
    i = j + 1
    if re.match("(..)/*", bond[i][0]).group() == re.match("(..)/*",
bond[i+1][0]).group() and re.match("(..)/*", bond[i][0]).group() !=
re.match("(..)/*", bond[i-1][0]).group():
        data = bond[i].tolist()
        final_data.append(data)
```

```
bond = pd.DataFrame(final_data, columns=["date", "code", "name", "close",
"high",
                                    "low", "opem", "pre-close", "variation", "variation-rate",
"volumn", "amount"])


# build the dataset
cpi = pd.read_csv("./data/cpi.csv")
cpi_data = cpi["cpi"][:len(bond)-1]
cpi_data = ((cpi_data[:-1].values - cpi_data[1:].values) / cpi_data[1:].values)
m2 = pd.read_csv("./data/money_supply.csv")
m2 = m2["m2"][:len(bond)-1].astype(float)
m2 = ((m2[:-1].values - m2[1:].values) / m2[1:].values)
bond_data = bond["close"][1:]
bond_data = ((bond_data[:-1].values - bond_data[1:].values) /
bond_data[1:].values)
data = np.array([m2, cpi_data, bond_data]).astype(float)


# correlation analysis
cov = np.cov(data, bias=1)  # covariance matrix
corr = np.corrcoef(data)  # correlation coefficient matrix
print("covariance matrix of M2, CPI and the index of China treasury bond：")
print(cov)
print("correlation coefficient matrix of M2, CPI and the index of China treasury
bond：")
print(corr)


# separate the dataset into training set and testing set
X = pd.DataFrame(data.T.tolist(), columns=["m2", "cpi", "bond"])
training, testing = dataset(X)
```

```
# reshape the testing set
training_X = np.array(training[["m2", "cpi"]].values)
training_Y = np.array(training[["bond"]].values)
testing_X = np.array(testing[["m2", "cpi"]].values)
testing_Y = np.array(testing[["bond"]].values)
X1 = X[["m2", "cpi"]]
x = np.array(X1.loc[[8]].values)
theta = lwl_regression(x, training_X, training_Y, LS=True)

hx = h(testing_X,theta)

# draw the data of the testing set
xxx = np.array(testing[["m2"]].values)
yyy = np.array(testing[["cpi"]].values)
zzz = testing_Y
fig = plt.figure()
ax = fig.add_subplot(111, projection = '3d')
ax.scatter(xxx, yyy, zzz, marker = '.')

    # draw the surface of the theta(parameters of the linear regression)
xx = np.array([-0.05, -0.05, 0.05, 0.05])
yy = np.array([-0.05, 0.05, -0.05, 0.05])
x1 = np.array([[-0.05, -0.05, 0.05, 0.05], [-0.05, 0.05, -0.05, 0.05]]).T
xx, yy = np.meshgrid(xx, yy)

zz = np.array(xx * np.float(theta[0][0]) + yy * np.float(theta[1][0]) +
np.float(theta[2][0]))
ax.plot_surface(xx, yy,
```

```
            zz.reshape(xx.shape), color="grey")
print(theta)
ax.set_xlabel('M2')
ax.set_ylabel('CPI')
plt.show()
```

# ДОДАТОК В ТЕЗИ КОНФЕРЕНЦІЇ БЕЗПЕКА СОЦІАЛЬНО-ЕКОНОМІЧНИХ ПРОЦЕСІВ В КІБЕРПРОСТОРІ

1. Тернетьєв О.М., Якубець А.О., Просянкіна-Жарова Т.І. Застосування SAS Enterprise Guide для виявлення зав’язків у предметно-орієнтованих даних/ Інформаційні технології: теорія і практика: Тези доповідей II Всеукраїнської інтернет-конференції здобувачів вищої освіти і молодих учених (4 квітня 2019 р., м. Запоріжжя) [Електронний ресурс] Редкол.: А.В. Бакурова, Г.Л. Козіна, М.В. Новожилова, С.А.Ус, Д.В. Широкорад. Електрон. дані. Запоріжжя: ЗНТУ, 2019. 106-107 с. 1 електрон. Опт. Диск (DVD-ROM); 12 см. Назва з тит. екрана. ISBN 978-617-529-231-0

# ДОДАТОК Г ТАБЛИЦЯ СТАТИСТИЧНИХ ДАНИХ

| **Country** | **CPI** | **GDP - per capita (PPP) (US$)** | **Unemployment rate (%)** | **Inflation rate (consumer prices) (%)** |
|---|---|---|---|---|
| New Zealand | 9.5 | 28000 | 6.5 | 4 |
| Denmark | 9.4 | 37600 | 6.0 | 2.8 |
| Sweden | 9.3 | 40900 | 7.5 | 3 |
| Singapore | 9.2 | 60500 | 2.0 | 5.2 |
| Norway | 9.0 | 54200 | 3.3 | 1.3 |
| Netherlands | 8.9 | 42700 | 4.4 | 2.3 |
| Australia | 8.8 | 40800 | 5.1 | 3.4 |
| Switzerland | 8.8 | 43900 | 3.1 | 0.2 |
| Canada | 8.7 | 41100 | 7.5 | 2.9 |
| Luxembourg | 8.5 | 81100 | 5.9 | 3.4 |
| Hong Kong | 8.4 | 49800 | 3.4 | 5.3 |
| Iceland | 8.3 | 38500 | 7.4 | 4 |
| Germany | 8.0 | 38400 | 6.0 | 2.3 |
| Japan | 8.0 | 35200 | 4.6 | 0.3 |
| Austria | 7.8 | 42400 | 4.2 | 3.5 |
| Barbados | 7.8 | 23700 | 12.0 | 9.1 |
| United Kingdom | 7.8 | 36600 | 8.1 | 4.5 |
| Belgium | 7.5 | 38200 | 7.7 | 3.5 |
| Ireland | 7.5 | 40100 | 14.4 | 2.6 |
| Bahamas | 7.3 | 31400 | 14.2 | 3.2 |
| Chile | 7.2 | 17400 | 6.6 | 3.3 |
| Qatar | 7.2 | 104300 | 0.4 | 1.9 |
| United States | 7.1 | 49000 | 9.0 | 3.1 |
| France | 7.0 | 35600 | 9.3 | 2.3 |
| Uruguay | 7.0 | 15300 | 6.0 | 8.1 |
| Estonia | 6.4 | 20600 | 12.1 | 5 |
| Spain | 6.2 | 31000 | 21.7 | 3.1 |
| Botswana | 6.1 | 16200 | 7.5 | 8.5 |
| Portugal | 6.1 | 23700 | 12.7 | 3.7 |
| Taiwan | 6.1 | 38200 | 4.4 | 1.4 |
| Slovenia | 5.9 | 29000 | 11.8 | 1.8 |
| Israel | 5.8 | 31400 | 5.6 | 3.5 |
| Malta | 5.6 | 25800 | 6.4 | 2.7 |
| Poland | 5.5 | 20600 | 12.4 | 4.3 |
| Korea (South) | 5.4 | 32100 | 3.4 | 4 |
| Dominica | 5.2 | 14000 | 23.0 | 3.5 |

| | | | | |
|---|---|---|---|---|
| Bahrain | 5.1 | 27900 | 15.0 | 0.4 |
| Mauritius | 5.1 | 15100 | 7.8 | 6.5 |
| Costa Rica | 4.8 | 12100 | 6.5 | 4.9 |
| Lithuania | 4.8 | 19100 | 18.4 | 4.5 |
| Oman | 4.8 | 26900 | 15.0 | 4.1 |
| Hungary | 4.6 | 19800 | 10.9 | 3.9 |
| Jordan | 4.5 | 6000 | 12.3 | 4.4 |
| Czech Republic | 4.4 | 27400 | 8.5 | 1.9 |
| Saudi Arabia | 4.4 | 24500 | 10.9 | 5 |
| Malaysia | 4.3 | 15800 | 3.1 | 3.2 |
| Latvia | 4.2 | 15900 | 15.4 | 4.4 |
| Turkey | 4.2 | 14700 | 9.8 | 6.5 |
| South Africa | 4.1 | 11100 | 24.9 | 5 |
| Croatia | 4.0 | 18400 | 17.7 | 2.3 |
| Montenegro | 4.0 | 11700 | 11.5 | 3 |
| Slovakia | 4.0 | 23600 | 13.5 | 3.9 |
| Italy | 3.9 | 30900 | 8.4 | 2.9 |
| FYR Macedonia | 3.9 | 10500 | 31.4 | 3.9 |
| Brazil | 3.8 | 11900 | 6.0 | 6.6 |
| Tunisia | 3.8 | 9600 | 18.0 | 3.5 |
| China | 3.6 | 8500 | 6.5 | 5.5 |
| Romania | 3.6 | 12600 | 5.1 | 5.8 |
| Lesotho | 3.5 | 2000 | 45.0 | 5 |
| Colombia | 3.4 | 10400 | 10.8 | 3.4 |
| El Salvador | 3.4 | 7600 | 7.0 | 5.1 |
| Greece | 3.4 | 26600 | 17.3 | 3.3 |
| Morocco | 3.4 | 5100 | 8.9 | 1.4 |
| Peru | 3.4 | 10200 | 7.9 | 3.4 |
| Thailand | 3.4 | 9500 | 0.7 | 3.8 |
| Bulgaria | 3.3 | 13800 | 9.6 | 4.2 |
| Jamaica | 3.3 | 9100 | 12.7 | 7.5 |
| Panama | 3.3 | 14300 | 4.5 | 5.9 |
| Serbia | 3.3 | 10800 | 23.4 | 11.2 |
| Sri Lanka | 3.3 | 5700 | 4.2 | 7 |
| Bosnia and Herzegovina | 3.2 | 8200 | 43.3 | 3.7 |
| Trinidad and Tobago | 3.2 | 20300 | 6.4 | 5.1 |
| Zambia | 3.2 | 1600 | 14.0 | 8.7 |
| Albania | 3.1 | 7800 | 13.3 | 3.5 |
| India | 3.1 | 3700 | 9.8 | 8.9 |

| | | | | |
|---|---|---|---|---|
| Swaziland | 3.1 | 5400 | 40.0 | 6.1 |
| Tonga | 3.1 | 7400 | 13.0 | 6.6 |
| Burkina Faso | 3.0 | 1500 | 77.0 | 2.8 |
| Djibouti | 3.0 | 2700 | 59.0 | 5.1 |
| Indonesia | 3.0 | 4700 | 6.6 | 5.4 |
| Mexico | 3.0 | 14800 | 5.2 | 3.4 |
| Algeria | 2.9 | 7400 | 10.0 | 4.5 |
| Egypt | 2.9 | 6600 | 12.2 | 10.2 |
| Moldova | 2.9 | 3400 | 6.7 | 7.6 |
| Senegal | 2.9 | 1900 | 48.0 | 3.4 |
| Vietnam | 2.9 | 3400 | 2.3 | 18.7 |
| Bolivia | 2.8 | 4900 | 5.5 | 9.9 |
| Mali | 2.8 | 1100 | 30.0 | 2.9 |
| Bangladesh | 2.7 | 1700 | 5.0 | 10.7 |
| Ecuador | 2.7 | 8600 | 4.2 | 4.5 |
| Guatemala | 2.7 | 5100 | 4.1 | 6.2 |
| Iran | 2.7 | 13200 | 15.3 | 22.5 |
| Kazakhstan | 2.7 | 13200 | 5.4 | 8.4 |
| Armenia | 2.6 | 5500 | 5.9 | 7.7 |
| Dominican Republic | 2.6 | 9400 | 13.1 | 8.5 |
| Honduras | 2.6 | 4400 | 4.8 | 6.8 |
| Philippines | 2.6 | 4100 | 7.0 | 4.8 |
| Syria | 2.6 | 5100 | 12.3 | 4.8 |
| Guyana | 2.5 | 7600 | 11.0 | 2.2 |
| Nicaragua | 2.5 | 3200 | 7.3 | 8.1 |
| Pakistan | 2.5 | 2800 | 5.6 | 11.9 |
| Azerbaijan | 2.4 | 10300 | 1.0 | 8.1 |
| Nigeria | 2.4 | 2600 | 21.0 | 10.8 |
| Russia | 2.4 | 17000 | 6.6 | 8.4 |
| Ukraine | 2.3 | 7300 | 7.0 | 8 |
| Kenya | 2.2 | 1800 | 40.0 | 14 |
| Nepal | 2.2 | 1300 | 46.0 | 9.1 |
| Paraguay | 2.2 | 5500 | 6.6 | 8.3 |
| Zimbabwe | 2.2 | 500 | 95.0 | 5.4 |
| Kyrgyzstan | 2.1 | 2400 | 8.6 | 16.6 |
| Equatorial Guinea | 1.9 | 19600 | 22.3 | 7 |
| Venezuela | 1.9 | 12700 | 8.2 | 26.1 |
| Haiti | 1.8 | 12700 | 40.6 | 8.5 |
| Iraq | 1.8 | 3900 | 15.0 | 5.6 |
| Sudan | 1.6 | 2800 | 18.7 | 18 |
| Turkmenistan | 1.6 | 7900 | 60.0 | 12 |